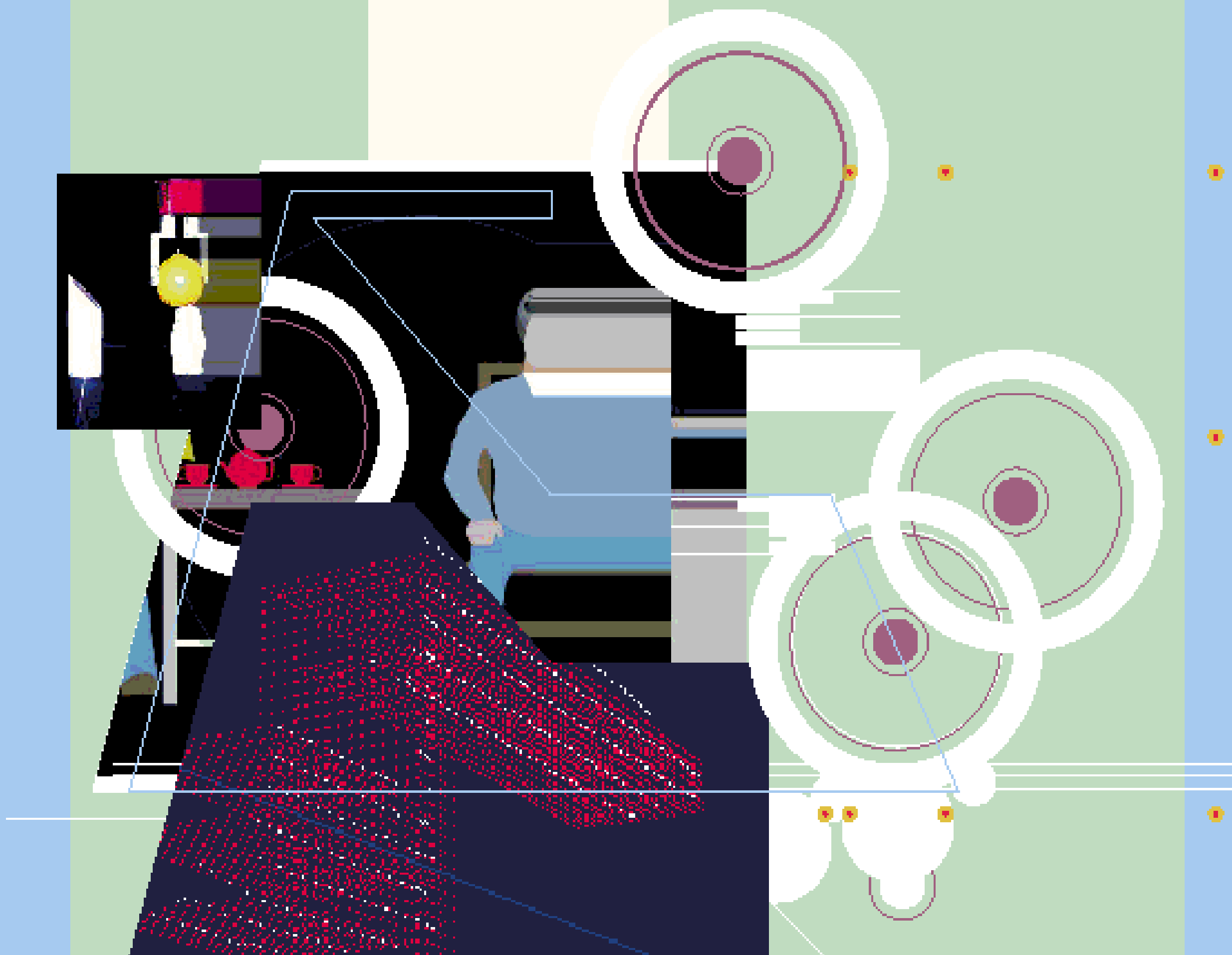
A Neuroprosthesis System Utilizing Optical Spatial Feedback Control



Spatial Integrated Systems, Inc.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
19-03-2004

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*
Aug 2003 – Feb 2004

4. TITLE AND SUBTITLE
A Neuroprosthesis System Utilizing Optical Spatial Feedback Control

5a. CONTRACT NUMBER
N00014-02-C-0384

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHOR(S)
Spatial Integrated Systems, Inc.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Spatial Integrated Systems, Inc.
7524 Standish Place, Suite 100
Rockville, MD 20855

8. PERFORMING ORGANIZATION REPORT NUMBER
SIS-R-202-04

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
Ballston Tower One,
800 North Quincy Street
Arlington, VA 22217-5660

10. SPONSOR/MONITOR'S ACRONYM(S)
ONR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
No restrictions

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The development of a neuroprosthesis system utilizing optical spatial feedback control is presented in this project final report. During this phase, the RobotEyesTM Functional Electrical Stimulation System (REFES) was developed as an intelligent vision based system. The system has capabilities in image capture and processing within a related small working environment. The working environment can be a fixed working table or a platform that satisfies varied conditions. The successful development of the REFES System during this phase is, in fact, a result of the application of a collection of advanced technologies that include real time image capture and processing, 3D surface reconstruction, 3D modeling and target recognition, camera calibration, robot control, intelligent trajectory planning, obstacle identification and avoidance, dynamic system identification, motion recovery and prediction, and position feedback control. The capabilities that REFES System provide is an effective user interface and optical spatial feedback controller for a neuroprosthesis for individuals with high tetraplegia resulting from high cervical spinal cord injury. It also provides for the command interface for rehabilitation robots that are commonly used by individuals with high tetraplegia, muscular dystrophy, amyotrophic lateral sclerosis (i.e., "Lou Gehrig's disease"), and other neurological or musculoskeletal disease.

15. SUBJECT TERMS
RobotEyesTM Functional Electrical Stimulation System (REFES) was developed as an intelligent vision based system.

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
				185	Ali Farsaie
					19b. TELEPHONE NUMBER *(Include area code)* 301 610 7965

Spatial Integrated Systems, Inc.

A Neuroprosthesis System Utilizing Optical Spatial Feedback Control

Final Report

March 19, 2004

Spatial Integrated Systems, Inc.
7524 Standish Place, Suite 100
Rockville, MD 20855
301-610-7965

Report No. SIS-R-202-04

TABLE of CONTENTS

ABSTRACT

The development of a neuroprosthesis system utilizing optical spatial feedback control is presented in this project final report. We report that - based on the work that has been done during this second phase- a neuroprosthesis system can be integrated with the REFES System based VZX system and a Functional Electrical Stimulator (FES) system.

During this phase, the RobotEyes™ Functional Electrical Stimulation System (REFES) was developed as an intelligent vision based system. The system has capabilities in image capture and processing within a related small working environment. The working environment can be a fixed working table or a platform that satisfies varied conditions. The integrated system is capable of all of the following:

1. Reconstructing certain types of 3D objects and the 3D scene encompassing the working environment
2. Gathering and processing 3D information and knowledge about the objects and the working environment
3. Understanding the gathered information and knowledge to allow monitoring of the changes of the working environment
4. Manipulating / utilizing the objects by controlling a robot arm with collision free movement
5. Tracking motion of a known object, such as a human hand or a robot end-effector.

The successful development of the REFES System during this phase is, in fact, a result of the application of a collection of advanced technologies that include real time image capture and processing, 3D surface reconstruction, 3D modeling and target recognition, camera calibration, robot control, intelligent trajectory planning, obstacle identification and avoidance, dynamic system identification, motion recovery and prediction, and position feedback control.

The capabilities that REFES System provide is an effective user interface and optical spatial feedback controller for a neuroprosthesis for individuals with high tetraplegia resulting from high cervical spinal cord injury. It also provides for the command interface for rehabilitation robots that are commonly used by individuals with high tetraplegia, muscular dystrophy, amyotrophic lateral sclerosis (i.e., “Lou Gehrig's disease”), and other neurological or musculoskeletal disease.

REPORT ORGANIZATION

This report is organized into the following sections:

1. **Introduction:** An introduction that explores the possibility of using optical spatial feedback control to develop a neuroprosthesis based on the current VZX system.
2. **VZX Imaging - manipulation arm integration:** This section describes overall system integration, including the hardware configuration, the 3D object and working environment design and integration and a system development between the REFES system and a simulator robot arm.
3. **Object Recognition, path planning, and navigation:** This section discusses how to design and implement an object operation based on the system set up described in the last section. Algorithms and methods development to guide the operation of the REFES are covered in this section.
4. **Arm movement assisted control:** This section focuses on motion tracking necessary to implement the ideas from the previous sections. Algorithms are described that allow tracking the motion of the robot arm or a hand model in order to provide position and orientation feedback for accurate FES control.
5. **User interface:** The development of a user interface was divided into a 2D graphic user interface (GUI) development and an assistant interface device application. Only the development of the GUI is discussed in this section.
6. **System requirements:** Brief introduction of identification of the range of motion neuroprosthesis system. Details of this discussion can be found in Appendix V of this document.
7. **Demonstration test 1:** This section discusses the accuracy of the 3D environment captured and understood by the REFES. The REFES accuracy results from tests performed on REFES systems at SIS and CWRU are outlined here.
8. **Demonstration test 2:** This section discusses the demonstration and test of a user interface and neuroprosthesis simulator arm tracking. The REFES tracking accuracy results from tests performed at SIS and a hand tracking demonstration at SIS are discussed here. The user interface demonstration discussion can be found in Appendix V of this document
9. **Conclusions:** The conclusions reached from the REFES development stated in this section and future work is discussed.

1. INTRODUCTION

The purpose of this project is to develop a neuroprosthesis system by integrating the RobotEyes™ technology based VZX system and a Functional Electrical Stimulator (FES) into a single functional system. FES is also called "neuroprostheses" and it acts as a substitute for the function of the paralyzed nervous system. FES works by electrically activating the nervous system distal to the injury to elicit coordinated contractions of the paralyzed muscles that provide useful function. The system is intended as an aid to individuals who have lost the use of basic muscle functions. In this report, the system is referred to as the RobotEyes™ Functional Electrical Stimulation (REFES) system.

1.1. FES SYSTEMS FOR INDIVIDUALS WITH HIGH TETRAPLEGIA

Individuals with high cervical spinal cord injury suffer from a condition referred to as high tetraplegia. These injuries are at the highest level of the spinal cord and leave those afflicted with extensive paralysis below the neck. Typically such individuals are left with volitional control of only the head, neck, and in some cases the ability for a shoulder shrug. Individuals with high tetraplegia are usually totally dependent on others for all aspects of care. Traditional rehabilitation procedures offer very limited options and result in limited functional improvement.

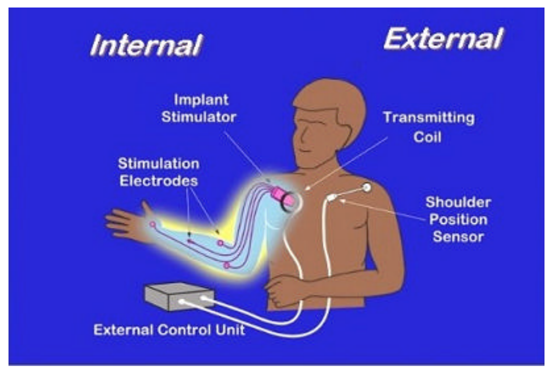


Figure 1.1: FES system consists of an external and internal sub-systems.

Neuroprostheses are systems that apply controlled electrical stimulation to paralyzed nerves and muscles to restore function. In an FES system, stimulating electrodes are implanted into patients' nervous system. The system consists of an external and internal sub-system as illustrated in Figure 1.1. The external sub-system consists of a control unit that generates electrical signals to the electrodes to either initiate or suppress movements of the paralyzed muscles. These systems can be used to restore different functions to individuals with a variety of different neurological disorders, although many applications to date have been for individuals with spinal cord injuries. Specifically, neural prostheses based on functional electrical stimulation have been deployed for restoring upper extremity function and a number of other functions. Specific to individuals with high tetraplegia resulting from high cervical spinal cord injury, several types of assistive devices can be used in conjunction with the retained movement function of the head and mouth to increase the independence. However, these assistive devices are difficult to control and are not currently portable enough for use in the community. Detailed discussion can be found in Details discussion can be found in Appendix V.

.

1.1.1. FES SYSTEM DEVELOPMENT IN CLEVELAND FES CENTER

The Cleveland Functional Electrical Stimulator Center is the worldwide leader in the development of FES systems. In addition to the work underway in the Cleveland FES Center to develop a neuroprosthesis for high tetraplegia, other groups have worked on this problem. An FNS system was used to restore function in an individual with C4 tetraplegia. The system attempts to restore movements by percutaneous stimulation of multiple muscles of the shoulder, elbow, wrist, and hand using stimulation patterns based on electromyographic (EMG) activity in able-bodied individuals. Pre-programmed sequences for different upper extremity activities are elicited by respiratory function (puff and sip). A balanced forearm orthosis was incorporated into the system to augment elbow flexion and shoulder stability and was identified as the most important factor in successfully utilizing their FNS system for functional tasks. An FNS system was also used to restore function in high tetraplegia. The system used surface electrodes that were held in place by an elastic sleeve. Splinting and the use of a sling-augmented voice controlled stimulation to the extremity. Two individuals with C4 level injuries have used the system to write and drink, and expressed the psychological benefits of seeing and feeling their arms move. Details discussion can be found in Appendix V.

1.1.2. CLOSE-LOOP MOTION CONTROL IS A KEY IN FES SYSTEM DEVELOPMENT

The implanted stimulators have already been developed and used in individuals with lower levels of spinal cord injury (and with less disability). The stimulating electrodes and the lead wires that connect them to the stimulator units have already been developed and are in wide use in other systems. The command interface is particularly critical because the neuroprosthesis for high tetraplegia requires the development of appropriate control algorithms to control multiple degrees of freedom of the arm and hand in individuals who have very few voluntary movements that can be used for control. A number of approaches have been proposed (see discussion of Task b below), but all are difficult and tedious because the user must continuously command the position of the arm as it moves to acquire an object and then somehow provide a separate command to open and close the hand around an object of interest. While accurate position control is normally not required for lower extremity FES, upper extremity functions, such as picking, putting and drinking, precise position and orientation of the FES-controlled arm require very precise position measurements. However, all existing clinical neuroprostheses operate as open-loop feed forward systems. i.e., the FES commands are generated based upon the known properties of the system and no automatic, sensor-based feedback is used to correct for errors due to external disturbances, fatigue, or other changes in the properties of the system. In high tetraplegia, the entire upper extremity is paralyzed, so voluntary correction for errors in the performance of the FES system will be very limited (i.e., perhaps just shoulder shrug). The number of functions to be simultaneously controlled by FES (hand, wrist, forearm, elbow, and shoulder) is simply too great for the user to be able to make command-based corrections in the performance of more than one or two of them. Previous FES systems devised for individuals with high tetraplegia have attempted to address the complexity of restoring many functions simultaneously by limiting the repertoire of restored functions. Pre-programmed stimulation sequences for a small number of activities, based upon the EMG patterns observed in able-bodied subjects, were stored in the controller. The user would evoke the performance of a particular activity by a single command and the motion was thereafter played out from beginning to end without user intervention. This approach has been used in a very small number of individuals and is not currently implemented in any users.

A closed-loop controller using the hand position and orientation tracking errors as feedback control input provides a solution to correct the FES patterns. Other projects in the FES Center are developing a feedback controller that automatically generates the stimulation sequences needed to restore a wide range of user-controlled arm movements while also providing feedback compensation control law for disturbances such as changes in load or fatigue. The feedback control law will correct the FES patterns to keep the endpoint location of the hand where desired and to maintain a desired hand orientation during movement so that, for example, the contents of a cup held in the hand do not spill while the cup is moved towards the mouth.

Using sensors of arm position in 3D space to provide feedback control has been considered and tested. In FES Center's near-term plans, sensors mounted on the forearm, upper arm, and trunk will provide feedback regarding the location of the hand in space and the orientation of the hand. While the sensor provides the feedback control, it has a number of disadvantages:

1. The user would be required to wear a large network of externally mounted sensors.
2. The neuroprosthesis would have to provide power to those sensors.
3. The user needs to put the sensors on each time the system is used.
4. There is no guarantee of the accuracy obtained from the orientation sensors because the sensors require an accurate transformation matrix and a highly repeatable location of the sensors across each use by the user or their caregiver.
5. Artifacts maybe present in the body-mounting orientation sensor signals related to soft tissue motion (i.e., relative motion between the underlying bone and the sensor due to muscle, fat, and skin properties).

Because of these disadvantages, it is very undesirable to install many external sensors on a human arm for positioning control.

1.2. REFES IS THE SOLUTION FOR MOTION CONTROL

In this phase, The REFES was developed to play two important roles in this neuroprosthesis. First, the system can provide a vision-based arm motion feedback signal needed to the closed-loop controller. Second, the system provides knowledge of the 3D workspace, including locations of the operational objects, trajectory calculation for acquiring the objects, and avoidance/knowledge of obstacles to guarantee safe operation of the robot arm.

The imaging component of REFES was developed from the VZX system. VZX is a SIS product with advanced 3D image capture and processing techniques. It automatically scans objects and, using digital imaging, creates 3D point clouds and a 3D model of the objects' space. It consists of a digital camera, a slider to move the camera, a stripe projector and a controller of the image capture. This is illustrated in Figure 1.2. The VZX system provides the ability to map an environment in three dimensions. By using the 3D environment information of a workspace, REFES system has been evolved to an intelligent vision system with knowledge of the 3D environment, functions to monitor the 3D environment and control of operations within the 3D environment. In other words, it utilizes the 3D information for real time navigation within this mapped 3D environment. For the neuroprosthesis system, REFES provides all 3D operation information, including locations, orientations, sizes of operating targets, positions, orientations and motions of a moving hand, dynamic operating path planning and operating environment monitoring. This information enables the control of hand operation under FES system.

Figure 1.2: VZX imaging system is a vision-based 3D image capture and processing system

In REFES, the REFES system has been designed and developed with a user interface that provides movement commands that are then executed by the feedback controller contained within the neuroprosthesis. REFES system also provides the vision-based motion feedback signals needed for the closed-loop control law. It predicts the hand motion one step ahead based on the previous observed motion. The prediction enables the controller to generate a compensation control law to overcome the current motion error and possible motion error one step ahead based on the feedback signal. The REFES system not only ultimately replaces these body-worn sensors by providing camera-based estimates of endpoint position and hand orientation, but also plans a moving trajectory to pick up objects avoiding any obstacles.

The general approach to a neuroprosthesis for high tetraplegia with the REFES system used as a component of this neuroprosthesis is illustrated in Figure 1.3. Two implanted stimulators produce needed contractions by passing current through the implanted electrodes into the paralyzed muscles (the "plant"). The external control unit provides power to the implanted stimulators via inductive links and provides the computational capacity needed to implement the feedback control algorithm. The REFES system becomes a combination of user interface and motion controller. It takes the user inputs and generates a sequence of commands to manipulate the motion of the human arm to perform the motion operations. A procedure of a simple operation can be described as:

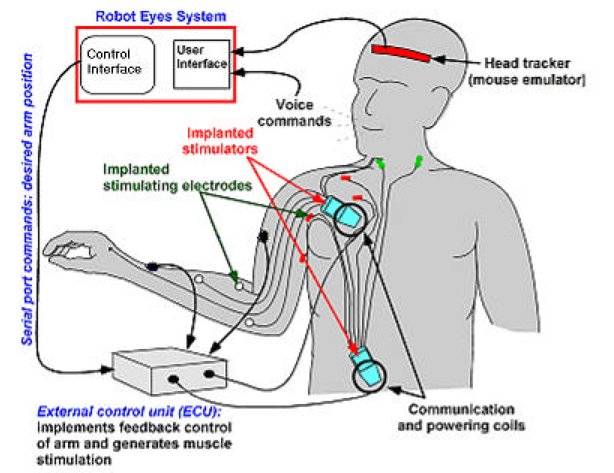


Figure 1.3: Schematic representation of RE integrated in to neuroprosthesis for an individual with high-level spinal cord injury.

1. The user selects an object and a destination on the REFES display screen;
2. The REFES system determines the object location, size and gripper pick up orientation;
3. The REFES system computes a trajectory that moves the arm to the desired object;
4. The REFES system sends the moving trajectory to the external control unit;
5. The REFES system monitors the motion of the human arm controlled by FES, sends position feedback control law the external control unit;
6. The REFES system sends control command to external control unit to avoid collision if any obstacle appears.

The Cleveland FES Center is currently involved in the development phase of the neuroprosthesis for high tetraplegia. No individual with high tetraplegia have yet been provided with a neuroprosthesis, although initial human implementation is scheduled in a two-stage procedure over the next 12-18 months. In the absence of paralyzed subjects with neuroprostheses, during this phase, the approach to developing human command interfaces, including one based upon the REFES system, has been done by using a robotic simulator. For the simulation purpose, a FES controlled human arm is simplified as a robotic arm. In a simulation of neuroprostheses, a robot arm with dimensions similar to the human arm is controlled by an able-bodied subject just as if it were their own paralyzed arm,

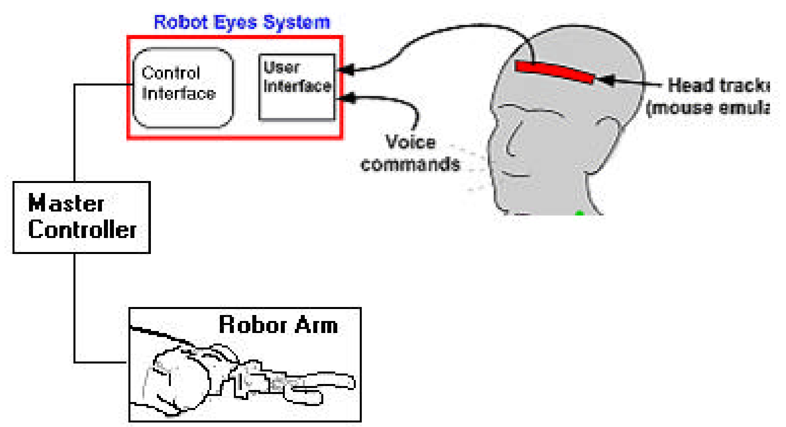


Figure 1.4: A robot arm is used to simulate of FES control motion.

an effective simulation of an individual with high tetraplegia. Such an approach allows us to rigorously evaluate potential command interfaces before we actually implement such a neuroprosthesis in an invasive and expensive surgical procedure. The robotic simulator developed during this phase is illustrated in Figure 1.4.

In summary, the primary advantage of the REFES system as a command interface for high tetraplegia is that the user need specify only the object he or she wishes to acquire via a simple visual interface. REFES system then automatically computes a trajectory from the current location to the desired location that avoids any obstacles and approaches the desired object in an appropriate manner. The advantages of using the REFES system can be concluded:

1. The REFES system provides knowledge of 3D workspace that the motion sensor doesn't have;
2. The REFES system provides operation trajectory planning that the motion sensor can not;
3. The REFES system provides obstacle identification and avoidance that the motion sensor can not;
4. The REFES system provides 3D workspace monitoring and management that the motion sensor can not;
5. The REFES system provides vision motion feedback so that the user will not be required to wear a large network of externally mounted sensors.
6. By using the REFES system, the neuroprosthesis does not have to provide power to those sensors.
7. The REFES system is separated completely from the user so that the user doesn't need to put sensors on each time when the system is used.
8. The REFES system provides much better position and motion tracking accuracy.
9. Artifacts in the body-mounting orientation sensor signals related to soft tissue motion (i.e., relative motion between the underlying bone and the sensor due to muscle, fat, and skin properties) will be completely avoided.

2. VZX IMIGING - MANIPULATOR ARM INTEGRATION

During this task, the system integration has been discussed and expanded from the development in Phase I to include trade offs between different data formats and communications protocols to achieve maximum processing efficiency. Higher processing efficiency has been developed in order to enable the system to work in real time. The integration planned for this phase was performed with a REFES system prototype upgraded through the effort from Phase I. Knowledge about the 3D objects in the environment was used to plan gripper application, path planning, and placement of moved objects. The error feedback from the arm was estimated and predicted and ready to be used to modify the movement of an arm passing through a planned path. Custom components necessary for this work were designed and built. The result was an application that demonstrates the desktop data capture, user interface and planning, and arm control needed in the final product. The development discussion of this task is divided into following sub-tasks:

1. Integration of REFES;
2. Design and fabrication of hardware (Robot simulators for human arm);
3. Interface between REFES and Robot simulators Design and fabrication of hardware;
4. Higher processing efficiency;
5. Integration of 3-D objects in the environment 3-D environment;
6. Knowledge about 3-D objects;
7. Manipulating 3-D objects;
8. Adjust current trajectory by using error feedback from arm tracking.

2.1. INTEGRATION OF REFES

The hardware system configuration and software structure design is discussed in this section. Hardware concept and a big picture of the logical system for REFES is discussed here, to provide an understanding of connections between the REFES system and FES and how they were designed and developed.

2.1.1. REFES HARDWARE SYSTEM CONFIGURATION

The REFES hardware system consists of an FES simulator and the REFES hardware system and a robot work environment. The FES simulator is a 6 degree-of-freedom (DOF) industrial robot with a controller and a PC. The designed hardware system of the REFES system consists of a PC, a VZX system, two 2D tracking cameras and a user interface of both head tracker and voice device. The computer required here must provide least a 1GHz processing speed and memory capacity of no less than 500M RAM. The system operates with Window 2000 or Window XP. The computer system was equipped with two IEEE 1394 PCI cards and two serial ports, providing communication with the two 2D tracking cameras and the FES simulator computer system. The VZX System connects to the REFES computer with a single firewire cable. The robot work environment consists of a robot working table and a set of experimental 3D

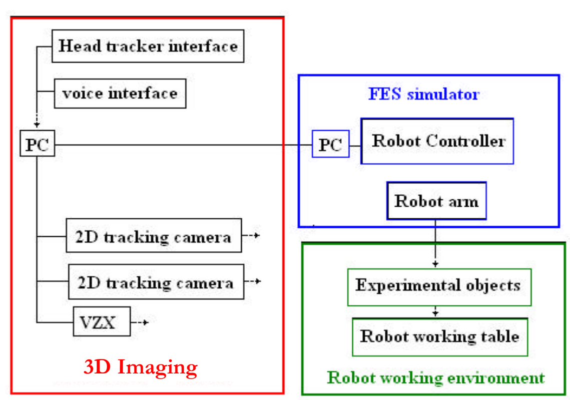


Figure 2.1: REFES hardware system configuration.

objects. The hardware system configuration is illustrated in Figure 2.1 and explained later during the demonstrations. The robot system was designed to be mounted on either the robot worktable or anywhere in order to enable the workspace of robot arm to match the workspace over the table. The workspace over the table was designed to be a sub space of the views of the VAX camera and two 2D tracking cameras.

2.1.2. THE REFES LOGICAL DIAGRAM

A REFES consists of the REFES system and a simulator robot of the FES system. It has been developed during this phase to demonstrate how a neuroprosthesis system can be integrated with a single system. The logical diagram of the REFES is illustrated in Figure 2.2. VAX camera and 2D tracking cameras are set up facing to the FES simulator robot arm and the robot workspace. Images contain vision information are captured by these cameras. While the VZX provides 3D information of the robot arm, its workspace and objects within the workspace, the tracking cameras capture real time 2D image sequences. Proprietary algorithms and functions have been designed and developed to process 2D real time image sequence. First, the motion tracking algorithms was developed to identify the motion of the robot arm and later, to identify motion of a hand model. The motion information from the robot arm and hand model is used to predict the next step moment and to provide feedback control signal for the neuroprosthesis system. Second, the obstacle identification algorithm was developed to provide obstacle information. The obstacle information is later used in dynamic trajectory planning. Finally, robot workspace monitoring algorithms have been developed to identify any new objects appear within the robot workspace. Together with the user recognition of the selected object, the 3D information provided by the VZX is used to recognize an object pick up orientation for the robot picking operation. The environment manager is designed to manage the robot workspace, including robot arm locations and moving trajectories, object locations, sizes, orientations and obstacle and new object locations, etc. There are two trajectory-planning algorithms that have been developed in this phase, the trajectory planning and dynamic trajectory planning. The trajectory planning is designed to provide an initial trajectory after a user operation command is given and before the robot arm starts the operation. The dynamic trajectory planning is designed to provide real time trajectory planning in case an obstacle appears within a trajectory in order to avoid any collisions. The function of the workspace design is to define a robot operating space. The workspace uses an environment manager to validate a trajectory. The user interface is designed to provide robot operation inputs such as selection of an object and destination of the operation.

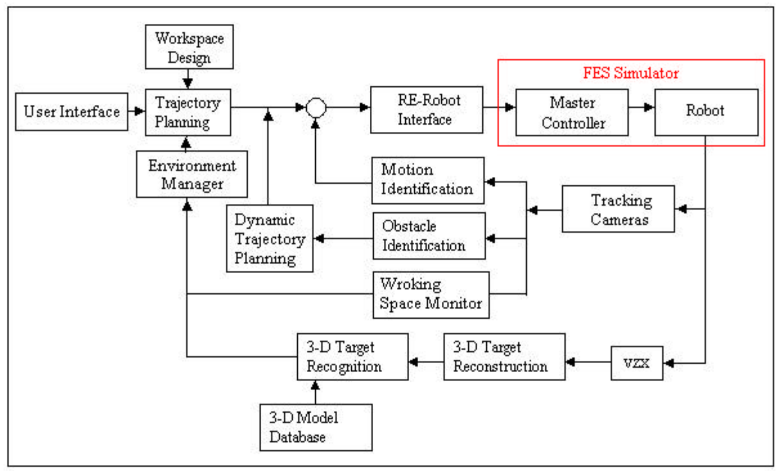


Figure 2.2: REFES logical diagram

2.1.3. A REFES WORKING SCHEME

Based on the REFES configuration, a typical REFES operation scheme can be described. The user selects an object on the REFES display screen either by using a head pointer (a mouse emulator) or by voice commands. The user also selects a desired destination point. The point can either be a 2D location from the graphical user interface or a given 3D position such as the user's mouth position. The REFES system proceeds as follows:

1. Decides which object was selected based on the user's inputs
2. Determines the object location, size and gripper pick up orientation by matching the selected object to an object database
3. Translates the user selected destination to a 3D position within the robot working-table if a 2D position is selected,
4. Computes a trajectory that moves the arm to the desired object while avoiding obstacles, and sends this trajectory via a serial port to the external control unit to provide the movement command.

For example, if the user specifies a coffee mug, the REFES system first calculates the 3D location by matching a set of 3D points that represent the single view of coffee mug to a complete model of coffee mug from the database. Second, the REFES system calculates a 3D destination,. The third, REFES system designs a trajectory that moves the hand to the mug handle. This approach could completely relieve the user of the burden of continuously controlling a trajectory through a cluttered workspace. The ability to recognize and acquire a wide range of items useful in everyday activities has the potential to significantly enhance the independence of the neuroprosthesis user and to decrease the amount of caregiver time needed each day (with substantial financial savings).

2.2. ROBOT SIMULATORS FOR HUMAN ARM

The selection of hardware system components and the design of their configuration are discussed in this sub-section. The selection of hardware system components includes selections of the robot arm, selection and design of robot gripper, design of robot controller, design of robot working table, design of VZX set up and its installation, selection of 2D tracking cameras, design of 2D tracking camera set up and installation, and selection and design of experimental object design, etc.

2.2.1. ROBOT ARM SELECTION

The Mitsubishi RV1A robot was selected as a simulator of the human arm for the REFES system at SIS and later, the Staubli RX60B robot was selected as the human arm simulator installed in CWRU. Photographs of both Mitsubishi RV1A robot arm and Staubli RX60B robot arm and their mounting set ups are provided in Figures 2.3. The reasons for using the RV1A and RX60B are because they both have six degrees of freedom (DOF) and their maximum reaching distances are 650 mm and 410 mm, respectively. A robot with 6 DOF can reach most positions in their workspace and are capable of simulations of an arm motion. Although a normal human arm operation can reach more than 6 DOF, the degrees of freedom an arm motion of an individual with high tetraplegia can operate is much less than the degree of freedom a normal human arm can achieve. The number of DOF the operation of a FES controlled neuroprosthesis system is not larger than 6 as expected. The maximum reach of 660 mm is almost identical to the average human reach of 650 mm. The Mitsubishi RV1A robot arm (left) was mounted on a

Figure 2.3: Mitsubishi RV1A robot arm (left) and Staubli RX60B robot arm (right)

robot working-table and Staubli RX60B robot arm (right) mounted under a base attached to a pose.

RV1A robot is currently installed on a robot working-table in a laboratory in SIS. The robot working-table is specially designed to provide a play form to hold operating objects as a simulation of a dinner table or book desk to a patient. With the working environment, its workspace consists of a semi ring with respect to its X-Y plan as shown in Figure 2.4A illustrates the mechanical drawings on the robot working-table.

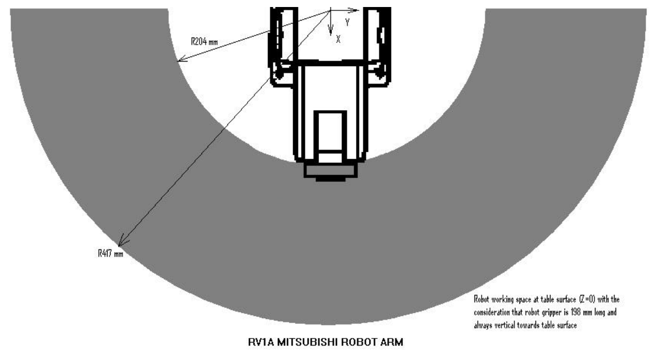


Figure 2.4A: A nominal workspace of the RV1A robot as installed.

The RV1A is a compact industrial robot developed for high accuracy operations. Its pose repeatability and distance accuracy is between –0.02 mm and +0.02 mm. The pose repeatability is defined as the value equal to the average of the maximum value and the minimum value of the group of attained poses with (+) or (-) added. The distance accuracy is defined as the distance from the teaching point to the point that is equal to the average of the maximum value and the minimum of value of the group of attained poses. The pose and distance accuracy is sufficient for all simulation operation experiments, including motion control, grasping operation, tracking and measurements during the tests. The mechanical measurement accuracy for the test purpose is from -0.1 mm to +0.1 mm. The object grasp accuracy will be discussed in sub-section 2.2.2. The maximum load capacity is 1.5 kg in which we considered would cover the weights of picking up object during the experiments. The maximum load capacity is defined as the mass with the flange posture facing downward at the +10 and –10 degrees limit. The RX60B robot was currently installed in an FES laboratory of CWRU, mounted in inverted position and at an appropriate height to approximate a human arm as shown in Figure 2.3. above. With its designed configuration and working table set up. Figure 2.4B illustrates the nominal workspace of the Staubli RX60B robot when mounted. Essentially, this is a large steel tube with mounting flanges at either end. The lower end was bolted to the floor using eight concrete anchors. A large aluminum block was bolted to the other end, with the robot cantilevered off the end of this block and hanging downwards. With the designed working environment and its mounted configuration, its nominal horizontal workspace provided is illustrated Figure 2.4B.

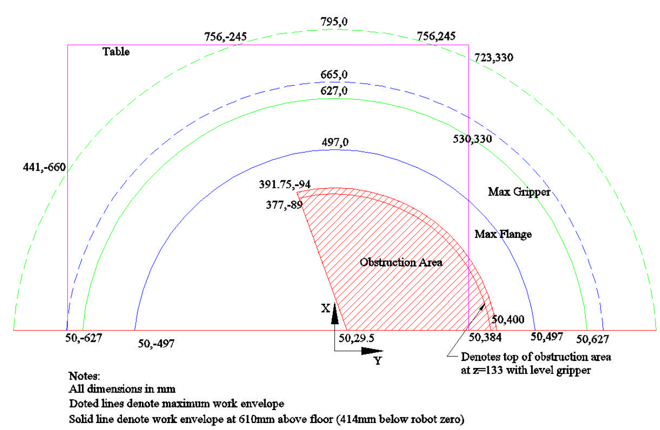


Figure 2.4B: Nominal workspace of the Staubli RX60B robot as mounted.

The area of this workspace is just slightly larger than would be expected to be accessible to an individual with high tetraplegia with an advanced neuroprosthesis that restores arm motions.

The RX60B has the maximum payload of 4kg at low speed and 2kg at full speed, both more than adequate for simulating the arm function of individuals with high tetraplegia who are expected to be relatively weak even with a high performance neuroprosthesis.

Because of the relatively large dimensions and mass (44 kg) of this robot and its unusual inverted mounting arrangement to simulate a human arm, a substantial mounting fixture was required.

2.2.2. GRIPPER DESIGN

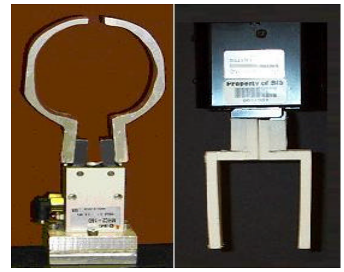


Figure 2.5: Gripper designed for both the RX60B (left) and RV1A robots (right).

The original design for gripper attached to RV1A robot end-effector is a typical robot gripper as shown in Figure 2.5. The gripper was selected based on the presence of a large robot operation space on a working table and a simple control of the pick up orientation. As shown in the figure, the robot arm is picking up an object (apple) apple from the working table. The robot gripper has a large amount of freedom for the pick up operations without any restrictions from the pick up environment because the gripper is hung down from the top to the bottom. The robot can always move the end-effector to the top of an object, select a suitable pick up orientation, rotate the gripper, move straight down to the object position, and grasp the object. On the other hand, the selection of the Staubli RX60B robot gripper was a finger-like gripper as shown in the left part of Figure 2.5. This finger design may be compared to a human hand with two large fingers or to a human hand with very small degree of freedom in finger operations. In this case there, only the simple operations of “open hand” and “close hand “ are available. The gripper was built to grasp large cups such as thermal insulated coffee mugs (~70mm diameter, see Figure 2.8). This is very much similar to the human hand with the FES controlled neuroprosthesis system. The left part of Figure 2.6 shows how the robot gripper picks up a cup. While this setup is close to the FES neuroprosthesis system, comparing the operations with the gripper and Mitsubishi RV1A robot configuration, the object picking up orientation control is much more complex because the pick-up orientation is restricted to a small degree of freedom.

Figure 2.6: Mitsubishi RV1A robot with pneumatic gripper mounted at the endpoint grasping an object.

2.2.3. ROBOT WORKING ENVIRONMENT DESIGN

A robot working table was designed and constructed for the Mitsubishi RV1A robot and was set up in the laboratory in SIS. Refer to Figure 2.7. below. The table height is 32" width, 36" long and 32" height. A Metal bracket was constructed as a table extension on which to mount the VZX system and the tracking cameras. This led to "sagging and vibration" problems when the VZX camera started capturing images. Experiments were taken to check out the vibrating problem. It was found that the vibration was caused by the weight of the VZX system. A solution for the vibration (noise) was examined. Some small wood gussets (braces) and necessary structural pieces were added. This successfully eliminated the vibrating. The gussets (braces) were very small so the visual appearance of the working table was not affected adversely.

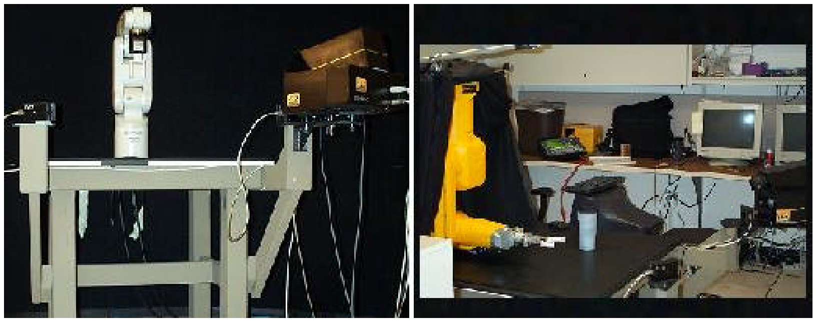


Figure 2.7: Robot working tables configuration with the Mitsubishi RV1A robot (left) and Staubli RX60B robot (right).

Mounting small leveling indicators directly to the camera mounting plate so that an individual can check to make sure nothing is moving or has moved did further improvement for the working table. In order to protect the table from damage due to possible collision of the robot gripper during incorrect operation, a black cover of rubber mat with non-glossy surface was used to cover the working table surface. This proved to also be beneficial in removing reflections and glare for the digital cameras. A black cover for the robot working table background was designed and placed in front of cameras and behind the robots.

The Working table is fastened to the column, which holds the robot and to the floor. This provides stability and guarantees that the orientation of the cameras to the robot remains constant. Mounting the legs to the floor eliminates the possibility of some one bumping the Working table thus effecting its orientation.

2.2.4. EXPERIMENTAL OBJECT DESIGN

Objects were designed for experimental operation purposes. While objects selected similar to those of future neuroprosthesis system operations, restrictions as to size and weight of the robot gripper grasp were also taken into consideration. All objects were smaller than 100 mm (upper size limit) and never larger than 50 mm (lower size limit). The weights of the objects are less than 1.5 kg. A number of objects were selected and used during this phase and are shown in Figure 2.8. For test measurement purposes, cylinder models with markers of angles at the bottom were designed and used to test operation positions and orientations. A coffee mug was used to test robot gripper grasping large cups. The choice of grasping relatively large objects was made for simplification purposes and was a first step toward manipulating smaller objects later on.

Figure 2.8: Object samples selected and used for robot operations, including identification, grasp, relocation.

2.2.5. ROBOT CONTROLLER

The Staubli robot includes a controller that can be used in the manner typical to an industrial robot. Because our goal is to emulate a person with a spinal cord injury using a neuroprosthesis and because we need to interface to the REFES system, a PC-based MasterController (MC) capable of very fast real-time control and capable of a wide range of interfaces was implemented. The MC receives robot joint angle trajectories from the REFES system, manipulates them as needed to emulate a paralyzed arm, and then sends appropriate commands to the Staubli robot. These joint angle commands are sent to the robot via a serial interface (115200 bps, 8N1). The joint angles are updated at 100 Hz for smooth operation. A checksum error checking routine has been implemented to account for dropped bytes in the data stream. Qualitative assessment of the communication between the MC and the robot was performed using transmission of simultaneous sinusoidal inputs to the joints and the arm tracks well.

2.2.6. ROBOT KINEMATICS

Kinematics equations for the robot arm were derived such that, for a given wrist location and orientation, the requisite 6 joint angles are calculated. However, these questions are not currently used in the MasterController as the REFES system provides joint angles directly. Note that we derived the inverse and forward kinematics equations for the Staubli robot out of necessity, since the proprietary source code was not available to the project for modification.

2.2.7. ROBOT INTERFACE WITH REFES SYSTEM

The primary factor considered for the requirements of interface between the REFES system and a robot manipulator was the long-term use of the REFES system in a neuroprosthesis system. The interface requirements pursued in this project are:

1. A robotic simulator with dimensions and joint motions similar to the human arm will be used as a proxy for the paralyzed arm of an individual with high tetraplegia. An able-bodied subject can control the robot arm just as if it were his or her own paralyzed arm - an effective simulation of an individual with high tetraplegia. Such an approach allows us to rigorously evaluate potential command interfaces, such as the VZX system, before we actually implement a neuroprosthesis for high tetraplegia in human user, deferring an invasive and expensive surgical procedure until we have high confidence that the neuroprosthesis will be successful.
2. The REFES system will work through the same controller hardware used by the rest of the neuroprosthesis. In particular, the REFES system will be compatible with a high-performance controller based on single board computers that is currently in the prototype stage in our laboratory.
3. Likewise, the interface between the REFES system and the neuroprosthesis must be implemented in the next generation neuroprosthesis software development tools under development in our laboratory. Specifically, this means that the REFES-to-neuroprosthesis interface must be implemented in Simulink (The Mathworks, Inc.) and executable under their “xPC Target” real-time environment.

After discussion with FES Center in CWRU, the following interface specifications were agreed upon: The processing of the REFES system is separate from the neuroprosthesis system. The communications between two processing will be complemented by the interface using a serial port.

This could be easily implemented on the REFES system and was accessible to the Simulink/xPC Target environment.

2.3. HIGH SPEED PROCESSING

High processing efficiency is a critical issue for the REFES system because the neuroprosthesis system is a real time system. Most of the system computations take place after the system starts operation. The only exception is that a 3D model database is created off line to match any target object during the operation. There are several considerations in the REFES 3D image capture process that can adversely affect system execution time: In order to provide sufficient 3D information for object 3D point generation, a sequence of images is taken by VZX running a camera over a slider. It takes about 40 second for this operation. To increase processing speeds the following can be done:

1. Reduce number of slider images captured by VZX from 101 to 51. This reduced the capture time from one minute to 30 seconds and reduced the processing time from one minute to 20 seconds.
2. Increase the processing speed of captured 3-D image points. The program to generate 3D point from captured image sequence was updated and the processing time was reduced from several minutes to a matter of seconds. The total image capture and processing time was reduced to one minute or less.
3. Increase processing speed of 3-D target recognition. Two algorithms are currently used: Two-Step recognition algorithm and One-Step recognition algorithm. The two-Step algorithm assumes a known object orientation and takes 5~7 seconds to recognize 1~5 objects. The one-Step algorithm is based on the Iterative Closest Point(ICP) algorithm. Its major advantage is that it is independent of the object orientation. It currently takes 4~30 minutes to identify 1~3 objects. Recognition speed of the One-Step algorithm depends heavily on the number of models in the database. A possible solution is to combine the two algorithms so that the speed can be reduced to less than one minute for a very large model database, for instance, the number of models is larger than 100.
4. Increase speed of real time image capture. There are two cameras used for the robot gripper tracking and obstacle detection. The capture speed of each camera has been increased to 30 image frames per second from the original 4~5 frames each second. This improvement made all real time operations possible, including obstacle identification and avoidance, end-effector or hand model motion tracking and workspace environment identification.
5. Increase real time image processing speed. Real time image processing provides motion capture and obstacle detection functionalities. The processing speed has been increased to more than 20 frames each second from the original two images per second.
6. Increase robot gripper operation speed. The robot gripper operation speed has been improved but still, it takes two seconds to make sure the gripper opens or closes properly.

2.4. THE 3-D ENVIRONMENT

A number of mathematical 3D sub-spaces were defined to create a 3D working environment. The workspace of the robot-working table is defined as all positions over the table. A robot working sub-space is defined as a sub-space of the intersection of the robot workspace defined by the manufactory and the robot table workspace. In other words, the robot working sub-space excludes any space under the working table or outside of the table. The camera view sub-space is defined as a 3D sub-space from where any 3D object can be projected into the 2D image captured by the camera.

The 3D environment includes selected 3D workspace and objects within the space. The 3D objects include manipulator arm, objects, robot working table, robot arm moving trajectories and possible obstacles and new objects. Configuration of the system to create the 3D working environments included in the demonstration test 1 and 2 sections of this document and are listed here:

1. Robot arm
2. Gripper
3. VZX 3D camera
4. 2D tracking cameras
5. Robot working table
6. Robot operating objects
7. Obstacles

2.4.1. LENS DISTORTION AND INTERNAL CAMERA CALIBRATION

The computer vision algorithms used in this project all use the concept of a perfect pinhole camera to model the relationship between the camera, imaging plane, and the physical world. The greatest weakness of the simple model is in the distortion introduced by a physical lens. Common lenses will all exhibit some degree of distortion, shifting the contents of captured images in nonlinear ways. Expensive lenses will reduce, but not eliminate, the problem. Therefore techniques for mapping this distortion are used, and these distortion maps are used to remove the distortion from captured images by performing a transform, which reverses the distortion process. Camera calibration is the process of determining all of the significant parameters of a camera's internal structure so that the relationship of the captured image to the real world is known. In the context of this project, lens distortion is the greatest unknown.

In the approach used here, lens distortion is modeled as centered about some point on the image, and radiating out from that center with radial and tangential components contributing to the image shift. Six parameters are used to characterize the distortion. Define (c_x, c_y) the coordinates of the principal point (center of lens distortion), (k_1, k_2) the radial components of distortion and (p_1, p_2) the tangential components of distortion. Then the mapping from the ideal (x, y) coordinate to the distorted coordinate can be solved by following equations:

$$\tilde{x} = x + x[k_1 r^2 + k_2 r^4] + [2 p_1 xy + p_2 (r^2 + 2x^2)] \quad (2.1)$$

$$\tilde{y} = y + y[k_1 r^2 + k_2 r^4] + [2 p_1 xy + p_2 (r^2 + 2y^2)] \quad (2.2)$$

where x and y are centered about the principal point, and r is the radius from the principal point.

To determine the six parameters, bundle adjustment is used. Bundle adjustment is a technique used to compute a maximum likelihood estimate of structure and position from image feature locations. In our case, a checkerboard target is imaged several times at various poses. The relation of the image features in the multiple images is formed into a non-linear system, and solved by using an iterative optimization process starting from a sub-optimal solution obtained by using linear methods.

The result of a successful bundle adjustment is a description of the principle point of the image, and a description of the radial and tangential distortion describing how much each pixel in the image must be moved to result in an undistorted image.

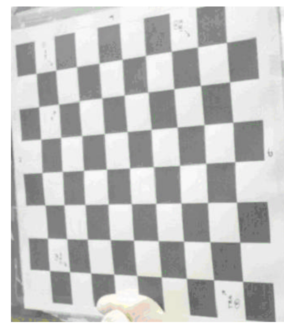


Figure 2.9: Image of a calibration target

Once the sum of the radial and tangential distortion offsets are applied to an image, the coordinates derived from the image can then be used in the traditional pinhole camera model. Another approach is to operate on distorted images and transform the resulting image coordinates to the ideal undistorted image, which may be computationally less expensive.

A check board is used as the calibration target and corners of the check board are identified and used as the principal points. Figure 2.9 shows the image of the check board. Part (a) of the Figure 2.10 illustrates the radial distortion offsets in pixels and part (b) illustrates the tangential distortion offsets in pixels.

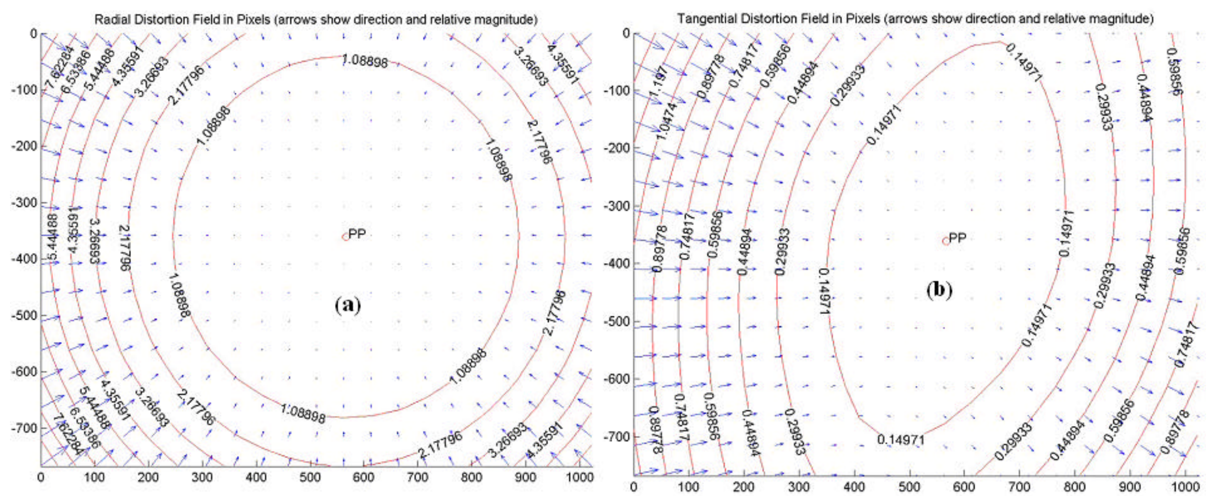


Figure 2.10: Image of a calibration target

2.4.2. EXTERNAL CAMERA CALIBRATION AND COORDINATE SYSTEM TRANSFORMATIONS

Camera Calibration is a necessary procedure in order to extract 3-D information from 2-D images. A number of methods have been proposed for solving the camera calibration problem in the past few years. These methods can generally be divided into two groups, Photogrammetric calibration and self-calibration. Photogrammetric calibration is done by observing an object whose 3-D geometry is known with good precision. The best known such method was used in TSAI, a project which automatically calibrates a camera, given two planes with a particular shape drawn on them (16 rectangles). The other method, self-calibration, is done by moving a camera in a static scene. The rigidity of the scene can be used to produce two constraints on the intrinsic and extrinsic parameters of the camera. Therefore, by obtaining pictures of the camera by different places, we can estimate the intrinsic and extrinsic parameters of the camera.

In this project, photogrammetric calibration was used to estimate the transformation from the 3D camera coordinate to robot coordinate and transformation from the 2D tracking cameras to the robot coordinate. A (4x4) matrix consisting of the extrinsic camera parameters represents the coordinate transformation while the intrinsic parameters were estimated by running a separate distortion correction program. The reason for using photogrammetric calibration to estimate the extrinsic parameter of the cameras is because the 3D geometry of certain parts of the robot can be measured with good precision. Particularly, the robot gripper is used to calculate the transformation matrix. The following reasons support the use of robot gripper as matching features for the camera calibrations:

1. It is located close to the focus points of the cameras (either 3D camera in VZX or the 3D tracking cameras);
2. It can be viewed whenever the system starts;
3. It can be tracked easily compare with other part of the robot body;
4. Its location can be calculated and measured easily, the position of the robot end-effecter can be calculated from the robot forward kinematics and the other locations of the gripper can be measured related to the end-effecter; and
5. There are no needs for any additional markers.

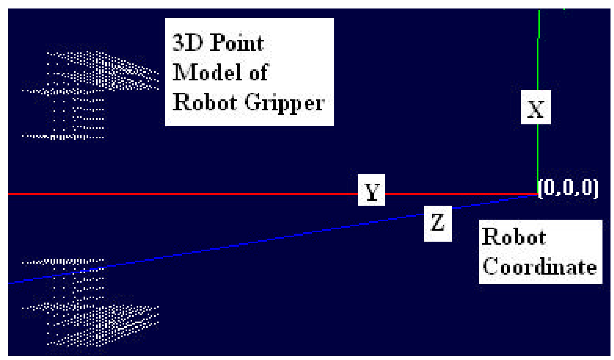


Figure 2.11: A “perfect” 3D point gripper model for RV1A robot system is created by measurement and used to calculate the transformation matrix from the VZX coordinate to robot coordinate

2.4.2.1. 3D EXTERNAL CAMERA CALIBRATION FOR RV1A ROBOT SYSTEM

A “perfect” 3D point gripper model for RV1A robot with respect to the robot coordinate system was created and used to calibrate the transformation matrix from the VZX 3D camera coordinate system to the RV1A robot coordinate system. The term “perfect” is used here to refer to a 3D point model generated by the VZX. Figure 2.11 shows the 3D point model of the gripper of the RV1A robot with respect to the robot coordinate system. Note that in the VZX display system, the coordinates are given in a left hand system. Experiments were performed to compare the calibration accuracy by using the ‘perfect’ model and the 3D point gripper model generated by the VZX. The results of the experiment shows that using the ‘perfect’ model can greatly improve the accuracy and consistency of the estimation. Although the VZX 3D point generation accuracy is smaller than 3%, its generation results may be inconsistent over time based on the environmental conditions, such as light, projection focus and so on.

The REFES system configuration is set up with the VZX 3D camera facing the robot system and provides that the robot workspace is a sub-space of the camera view. The transformation matrix from the VZX camera coordinate to the robot coordinate can be calculated from two projection models easily because there is not projection involved. Let $P_c = \begin{bmatrix} x_c & y_c & z_c \end{bmatrix}'$ be an object point within the camera view space (with respect of the camera coordinate) and $P_r = \begin{bmatrix} x_r & y_y & z_r \end{bmatrix}'$ be the object point in the robot workspace (with respect to the robot coordinate), then the relationship between P_c and P_c can be represented as:

$P_c = MP_r$ (2.1)	Where
	$M = \begin{bmatrix} R & T \\ 0_3^T & 1 \end{bmatrix}$, (2.2)
	$R = \begin{bmatrix} r_0 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix}$, (2.3)
and	$T = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}'$. (2.4)

The transformation matrix M maps point P_c to P_c. Matrix R is rotation matrix and vector T is translation vector.

When 3D point of the robot sense is generated by the VZX, it is used to match a set of points taken from 'perfect' model. In this project, the well known iterated closest point (ICP) matching algorithm is used to find the optimal matching. Figure 2.12 shows the gripper model and the gripper with respect to the camera coordinate in the left of the figure and shows the model gripper and the robot gripper with respect to the robot coordinate after the ICP matching program was run. Then the robot coordinate system was transformed by the transformation matrix M. The matrix is found by methods applying the ICP matching algorithm

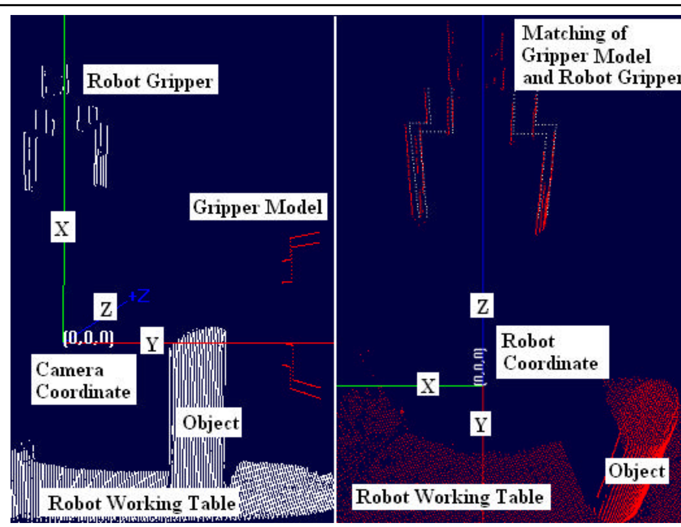


Figure 2.12: The robot gripper and perfect gripper model before (left: with respect to the camera coordinate system) and after (right: with respect too the robot coordinate system) the marching.

2.4.2.2. 2D TRACKING EXTERNAL CAMERA CALIBRATION FOR RV1A ROBOT SYSTEM

While the transformation matrix from the VZX coordinate to the robot coordinate system is done by matching a set of the robot gripper 3D points with a given set of 'perfect' gripper model, the transformation from the 2D tracking camera coordinate system to the robot system can not use the same method because the 2D tracking cameras have no knowledge of the 3D location of the robot. 2D tracking camera calibration was done by matching a 2D projection of a robot gripper with the image the tracking camera captured. The matching program used is the same program used in 3D camera calibration in section **2.4.2.1**.

2.4.2.3. VZX 3D EXTERNAL CAMERA CALIBRATION FOR RX60B ROBOT SYSTEM

A test cylinder object was used to estimate the transformation matrix from the VZX coordinate system to the RX60B robot coordinate system after the gripper matching method used at SIS failed in a test at CWRU. Large errors occurred when a "perfect" model of the finger like gripper was created and used to match the gripper 3D points generated by the VZX. This is because the gripper 3D points generated by the VZX had a large error compared with the perfect gripper model.

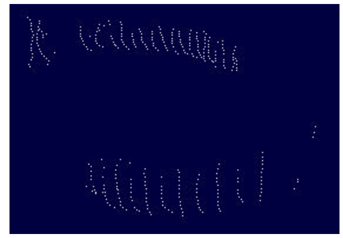


Figure 2.13: Reconstructed 3D surface of RX60B gripper is noisy.

Figure 2.13 shows a set of gripper 3D points generated by the VZX. It can be seem that the 3D points on the projection stripe are very noisy and do not form a surface of the gripper. While VZX generates very accurate 3D points for larger objects, and for small object such as gripper used in RV1A robot, object with a small irregular surface similar to the gripper in RX60B cannot be reconstructed with a high accuracy. Fortunately, under the assumption of the use of gripper, small objects are not used because the closed position of the gripper cannot hold on any objects smaller than 150 mm in diameter.

A cylinder model was originally designed for experimental and test purposes. It turned out to be a perfect calibration object. It was held by the robot gripper and placed at the robot home position. A set of 3D points generated by the VZX is then used to match with the 'perfect' model of the cylinder. The matching transforms the set of the 3D points of the cylinder with respect to the camera coordinate to the 'perfect' cylinder model with respect to the robot coordinate. The transformation is then used to transform to camera coordinate to the robot coordinate. Figure 2.14 illustrates the matching result with respect to the robot coordinate system.

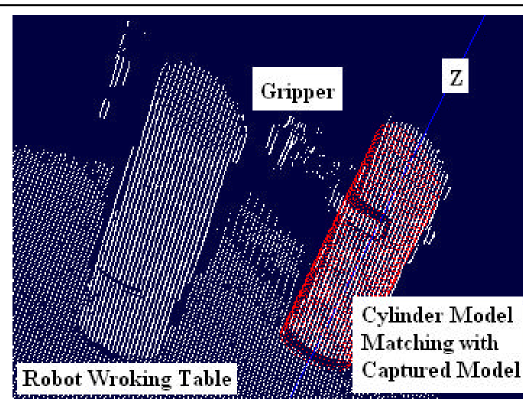


Figure 2.14: A cylinder model was used to estimate the transformation matrix from the camera coordinate to the robot coordinate system

2.5. REAL-TIME CONTROL NEUROPROSTHESIS SOFTWARE DEVELOPMENT

In order for the REFES to be independent from the application platform (e.g., either an FES system or a robot manipulator), a set of commands were defined for generic functions such as trajectory description, system initialization and shutdown. The Figure 2.15 below details these commands. The program then generates only these commands to communicate with the application platform and to control its actions. Therefore, for each actuating system, a separate program is required to translate such generic commands into the control language specifically for that system. This program is called RobotManager.

Function	Command	Reply	Remarks
Initialize and start up robot	S\n	R\n	Incl. open robot com port and turn servo on, etc.
Shut down robot	D\n	R\n	
Robot homing	H\n	R\n	Robot moves to its home position
Clear path	P\n	R\n	Clear any uncompleted path
Begin a robot path define	B\n	R\n	Describe a path by joint angles at each step
End a robot path define	F\n	R\n	End of the path description
Open robot gripper	O\n	R\n	
Close robot gripper with a grasping force	C\n +000.000\n	R\n	Reply only once after the force parameter being received
Move robot joints	M\n +000.000+000.000 +000.000+000.000+000 .000+000.000\n	R\n	Reply only once after all 6 joint parameters being received
Execute robot path	E\n	R\n L\n	Path defined between "B\n" and "F\n"; "L\n" is replied after motion being completed
Pause robot movement	A\n	R\n	Stop robot immediately, wait for further instruction
Resume robot movement	U\n	R\n	Resume the paused movement
Stop robot movement	A\n	R\n L\n	Stop robot immediately, clear any unfinished path, reset robot for further instruction. "L\n" is sent after robot being stopped and reset.
Get robot joint angles	G\n	G\n +000.000+000.000 +000.000+000.000+000 .000+000.000\n	Inquire current robot joint angles, in degrees.
Get robot gripper pose	X\n	X\n +000.000+000.000 +000.000+000.000+000 .000+000.000\n	Inquire current robot gripper pose (x, y, z, roll, pitch, yaw). x, y, z are in mm, angles are in degrees.
Send robot gripper pose	T\n +000.000+000.000 +000.000+000.000+000 .000+000.000\n	R\n	Provide current robot gripper pose for feedback control. Reply only once after all 6 parameters (x, y, z, roll, pitch, yaw) being received. x, y, z are in mm, angles are in degrees.

Figure 2-15: Robot Arm Generic Commands

RobotManager was developed in the C++ computer language for controlling a Mitsubishi industrial robot, RV-1A. It receives commands from the REFES program via serial interface (115200 bps, 8N1), and sends the translated instructions in robot language through another serial port (9600 bps, 8N1) to the robot controller. Figure 2.16 shows a block diagram of the communication and control

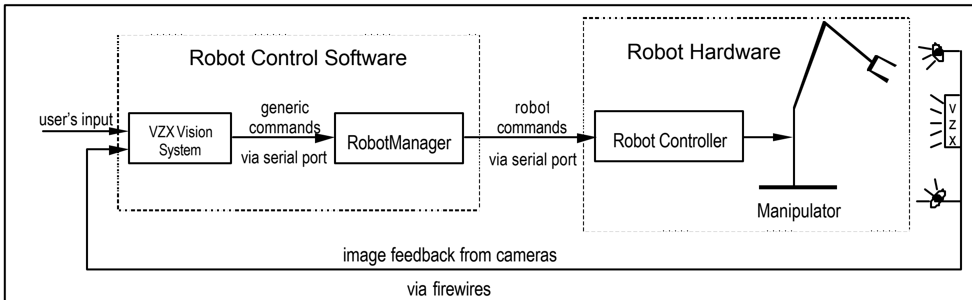


Figure 2.16: Block diagram of the robot system.

structure. Additionally, RobotManager also provides feedbacks to REFES for necessary robot conditions, such as whether the planned motion has been finished.

The robot commands shown in Figure 2.17 are used by RobotManager to translate all the generic commands given by REFES:

Command	Function	Command	Function
1;1;OPEN=NARCUSR	Open communication port	1;1;RSTPRG	Reset program
1;1;CLOSE	Close communication port	1;1;NEW	Start a new program
1;1;CNTLON	Enable robot operation	1;1;FDELTEMP	Delete file "TEMP"
1;1;CNTLOFF	Disable robot operation	1;1;LOAD=TEMP	Create file "TEMP"
1;1;EXEC SERVO ON	Turn motor servo on	1;1;EDATA5 J0=(0,0,90,0,90,0)	Define J0 on Line 5
1;1;EXEC SERVO OFF	Turn motor servo off	1;1;EDATA100 CNT 1,50,50	Define continuous move
1;1;VALM_SVO	Check motor servo status	1;1;EDATA110 MOV J0	Move robot to J0
1;1;OVRD=30	Set motion speed	1;1;EDATA999 HLT	Write "HLT" on Line 999
1;1;STATE	Check robot status	1;1;SAVE	Save program in "TEMP"
1;1;STOP	Stop robot movement	1;1;PRGLOAD=TEMP	Load program "TEMP"
1;1;EXEC HOPEN 1,63,20,.25	Open robot gripper	1;1;RUNTEMP	Run program "TEMP"
1;1;EXEC HCLOSE 1,63,20,.25	Close robot gripper	1;1;VALJ_CURR	Check current joint angles

Figure 2-17: Robot commands

Note that direct MOVE commands from RobotManager would require robot to stop at the end of each step along the trajectory. To achieve a smooth and continuous motion of robot end-effector, the entire robot trajectory has to be written into a temporary program in the robot controller. This is done by issuing "1;1;EDATA*** [command]" to the robot controller, where *** is the line number of the [command]. A typical program for the robot controller is listed below in Figure 2-18.

```
5       J0=(0,0,90,0,90,0)      % Define a step along robot trajectory, joints are in degrees
10      J1=(90,0,90,0,90,90)    % Define another step
100     CNT 1,50,50             % Define continuous motion for robot end-effector
110     MOV J0                  % Move robot to J0
120     MOV J1                  % Move robot to J1 continuously
999     HLT                     % Halt all robot operation
```

Figure 2-18: Typical program for robot controller

2.6. INTERFACE BETWEEN VZX VISIONING SYSTEM AND ROBOT SIMULATORS

With the specification that requires the REFES system working in a transparent manner with the software environment of the neuroprosthesis, it was determined that an interface between VZX visioning system and the neuroprosthesis MasterController should be designed. This design separated the development of VZX vision system, which includes both the 3D VZX and the 2D camera systems, from the neuroprosthesis control system and limit the connection of the two systems to a set of communication commands. Under the interface design, the VXZ system provides 3D environment based on 3D spatial information of objects captured by digital cameras and operations within the 3D environments based on user inputs and communicates with an objective of a robot arm as its external executive device by the commands. Figure 2.19 illustrates the logical relationship between the VZX visioning system and the neuroprosthesis system.

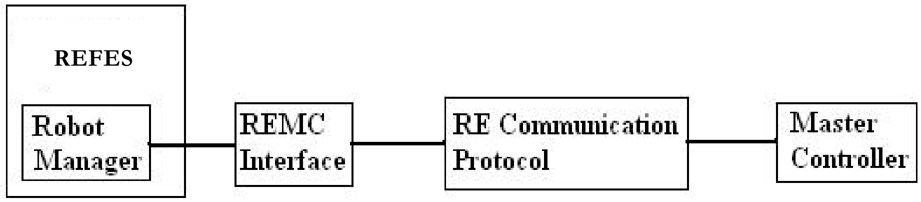


Figure 2.19: Interface between REFES and the neuroprosthesis system.

A logical object of RobotManager is created to represent the interface. A set of generic robot-independent control commands, called RE Commands, was designed. These commands are based on the 3D information of robot workspace and instructions from the user's interface. RobotManager interfaces with a robot arm by communicating with a logical object MasterController. MasterController is capable of real-time motion control of the robot arm. It receives and translates of REFES commands into robot motion control commands and sends required feedback to REFES. More details of the interface design can be found in Appendix II.

2.7. INTEGRATION WITH ROBOTMANAGER

The REFES system generates a set of generic c shown in Figure 2-17, ommands, and communicates with RobotManager via a standard serial port at 115200 bps. This section describes the interface protocol used for this communication.

All the generic commands are sent from VZX visioning system to RobotManager by using single ASCII characters. Each character corresponds to a specific function for the robot, e.g. as "M" for "move joints". All parameters are also in ASCII form and follow the format ±ddd.ddd, where "d" denotes a single digit number in the range 0-9. These ASCII strings are then converted into decimal values for further processing. A new-line return, "\n", always follows at the end of each command and at the end of a complete set of parameters. A reply has to be sent back to REFES following each command. The table below lists all the commands used between REFES and RobotManager, in which all parameters are given in "+000.000" for illustration purpose.

3. OBJECT RECOGNITION, PATH PLANNING AND NAVIGATION

The ability to visually guide the limb to or around the object and finally to pick up the object was an important stage to this development. The ultimate application to a paralyzed limb was considered in selecting an approach. During this sub-task, the robot working-space was designed to ensure the robot operation and path planning properties of an operating object - include the pick up orientation-were analyzed and defined. Path planning was designed and developed enabling the robot to move close to a pick up position. Gripper operation was defined and improved to perform pick up operations. Algorithms and routines to achieve these working tasks were developed. Significant software to implement the algorithms was developed, integrated and tested on the REFES system. Enhancements to tracking and path planning were performed.

3.1. ROBOT WORKSPACE AND OBJECT UNDERSTANDING

During this task, a robot workspace was defined based on the robot arm set up and its working environment. A robot workspace can be defined from the robot manufacturer configuration as simply all points the robot end-effector can reach. But in this project, a working environment was designed as a sub space over a robot's working table and under a level close to a human user's mouth. The robot workspace is therefore defined as an intersection of three sub-spaces, the robot manufactory workspace, sub-space over the table surface and sub-space lower then a distance of 400 mm from the table. All experimental objects and operations were limited to lie/occur somewhere within the robot workspace.

The operating objects were defined by a set of properties. All operating objects and their data are stored in a model database. The properties of these objects include the identifications, sizes, 3D locations, pick up orientations, and a full view of the 3D points. These properties were designed to enable related operation with the object. For instance, the REFES uses the name of an object to identify an object, uses the full view of 3D points to determine the location and orientation, and uses its size and pick up orientation to design a gripper approach and pick up operation.

A simple description of how the object properties are used in an operation is:

1. After a user selects an object name from the object list using the user interface, the system passes the name to the model database and searches for the properties of the object by the name.
2. The full view 3D points of the object then are used to match with the one-view 3D points of the object in the scene.
3. The matching results in the object location and orientation with respect to the robot coordinates.
4. The location and orientation information are then sent to the path planning procedure to generate a trajectory that controls the motion of the robot gripper to approach the pick up position.

The arm localization in this task is done by measurements of the robot arm configurations and calculation of joint variables generated by the robot controller. There are several reasons for doing this:

1. The joint positions of the REFES are properly known
2. It is difficult to identify the motion of different joints of the REFES without any markers
3. Even if it is possible to track the join position, the additional tracking computation time will significantly slow the tracking speed.

The arm's initial 3-D position can be localized by the known robot arm home position. In fact, the initial 3D location of a FES controlled "neuroprostheses" system is equivalent to the home position of a robot simulator arm. Using robot arm forward kinematics, join positions can be calculated.

It can be difficult to track the joint positions of a REFES because the position cannot be determined especially when clothes cover the human's arm. The features and outlines of the clothes can be changed when the arm moves. Some joints can be covered by another other part of the arm so that the system can loses the tracking.

To localize the joint positions from the robot joint measurements, robot arm configuration data is initialized when the system starts. Together with these data, a set of joint variables is used to calculate position of the joints at any given time. In this phase, a link between two joints then is used to estimate the 3D sub-space of the link within the robot workspace by calculating a sequence of spheres with a radius equal to the dimension of the arm and the centers along the link.

3.2. PATH PLANNING AND NAVIGATION

Path planning is one of the most challenging problems for successful applications of robots in many areas [1]. During the last two decades, research has been conducted extensively on various path-planning topics. The main objective is to find a collision-free path for a robot in the presence of obstacles either on-line or off-line with either fixed or moving barriers [2-4]. Whereas some research has focused on the path optimization problem in terms of energy consumption, moving duration or smoothness [5-7].

In general, there are two categories in designing a robot path: namely global and local techniques. In global solutions, a complete map of the robot environment is known in advance, and most of the efforts have been made in finding an artificial potential field that guarantees global convergence without local minima [8]. For this, a Configuration space, or C-space, needs to be constructed and the potential field has to be generated [8, 9]. These normally demand large amount of calculations, and consequently prevent the applications of the potential-field approach in real-time cases with dynamic environment. Other global methods, such as skeleton and cell decomposition [10], have similar difficulties in implementation for on-line solutions [2]. The local techniques, on the other hand, do not require the world model, but use only a small section of the robot environment and nearby obstacles for decision making. The path is determined by checking certain constraints, such as forbidden regions or penalty functions. These techniques have an advantage over global ones for having low computational complexity [11]. However, they suffer from the possibility of being trapped in local minima.

In this project, an effective 3D path planning method was developed for multi-degree-of-freedom robots in such environments. This method is based on a matrix representation of the robot workspace, and requires simple on-line calculations, thus it enables real-time obstacle-avoidance trajectory design.

Based on Section 3.1, the robot workspace is identified and all objects in this space are captured and recognized. Now it is necessary to represent them in a mathematical form to facilitate the robot path planning. To do so, a 3D matrix is used in which every element corresponds to a small cubical volume of the robot workspace (e.g., 1 cm^3). Four possible conditions are assigned to each element, which are Reachable-Free (RF), Unreachable-Free (UF), Reachable-Occupied (RO), and Unreachable-Occupied (UO). The conditions of reachable or not are determined by the robot kinematics, so for a given robot, these initial conditions can be assigned off-line at a preprocessing stage or read from a pre-established file. While, the conditions of free or occupied are based on the objects information, and must be assigned on-line and updated in real-time. By having the initial robot workspace as well as the position and the size of each object, this process can be achieved with minimal computer resources.

Before constructing any objects in the robot workspace, their sizes need to be enlarged to counterbalance the space occupied by the robot gripper and/or the object being held in the gripper, so that they can be shrank to a point. Then the desired trajectory is planned by moving this point throughout the workspace. Also, when multiple objects exist, they are distinguished from each other by using index numbers to avoid any confusion and unnecessary trial-and-error process during path planning.

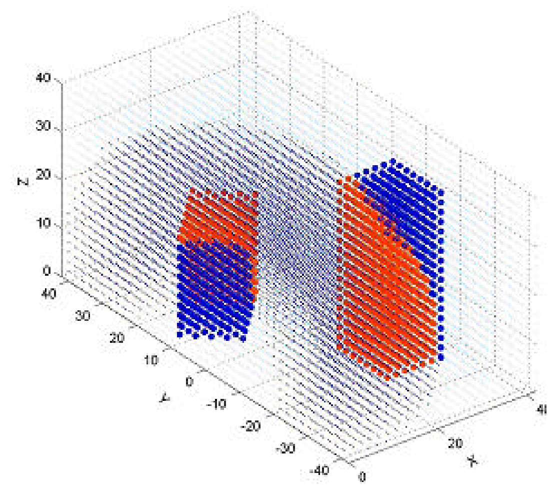


Figure 3.1. Robot workspace with two objects (in cm)

An example of the workspace of the RV-1A Mitsubishi industrial robot with two objects is illustrated in Figure 3.1. The small dots in the figure outline the point the gripper can reach within the robot workspace, and the zero level of the Z-axis indicates the height of the table surface with respect to the robot coordinate. The large dots denote the objects, both of which are partially outside the robot reachable area.

Once the robot workspace is depicted with a 3D matrix and all the objects are constructed, the robot path planning becomes relatively simple. In this project, we design a trajectory in the robot end-effector space based on three primitive paths, namely 'pre-pick path', 'pick-to-put' path, and 'home' path. In the pre-pick path, the robot starts from its home position and moves to the location of a user-selected object with a proper pick-up orientation identified in the next section. Then in the pick-to-put path, the robot lifts up the grasped object vertically from the table and carries it horizontally over to a user-specified target location for placing (or to a pre-defined mouth location for drinking and eating). At the end, the robot moves back to its home position.

Within each primitive path, sub-steps are interpolated to verify whether the path is free of obstacles by checking the corresponding elements of the 3D matrix. Upon a detection of an obstacle, two possible detours are considered to avoid a collision based on human's natural reaction: 1) to lift the arm higher to pass over the obstacle, or 2) to retract the arm closer to the body to get around the obstacle. At this time, the location and the size of the obstacle is known from its index number, thus a quick one-step path correction can be achieved.

Here, we assume that the robot environment is not congested with too many objects. This is reasonable since for a real patient with high-level disability, only necessary objects would be presented on his/her table. With this assumption, the possibility for a path being trapped in a local minimum is greatly reduced, and thus global solutions are not emphasized in this study. If a dead-end is reached, mostly it is because of an unsolvable case, so the user may have to initiate a removal of the obstacle to a new location in a separate operation.

3.3. DETERMINING OBJECT ORIENTATION DETERMINATION AND HAND NAVIGATION

In order to pick up an object correctly, such as the cube toy shown in Figure 3.2 photo (a), not only its location needs to be detected, but also the orientation has to be recognized so that the robot gripper can approach the specified object from an appropriate angle. Note that the 3D-point clouds

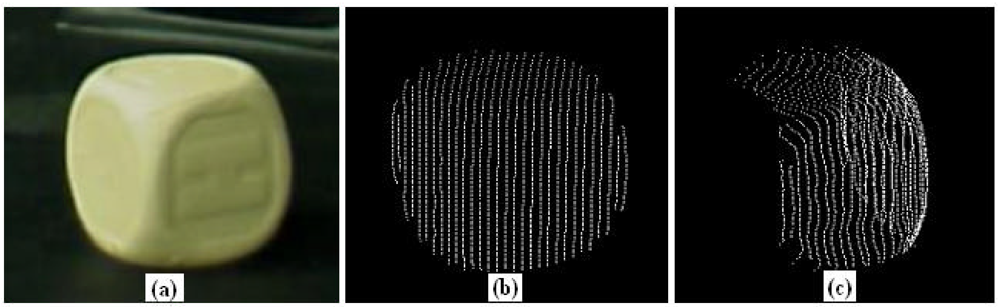


Figure 3.2: A cube with partial spatial information from VZX capture: (a) a scene of a cube toy, (b) 3D point cloud in the capture angle, (c) partial information of the cube toy

generated in a single capture by the VZX system provide just partial information of the objects since only one side of their view can be obtained, as shown in Figure 3.2 photos (b) and (c). To determine the location and orientation of an object, a matching of the partial view of the object with its 'perfect model' has to be conducted. This perfect model also consists of a set of 3D points but has a full

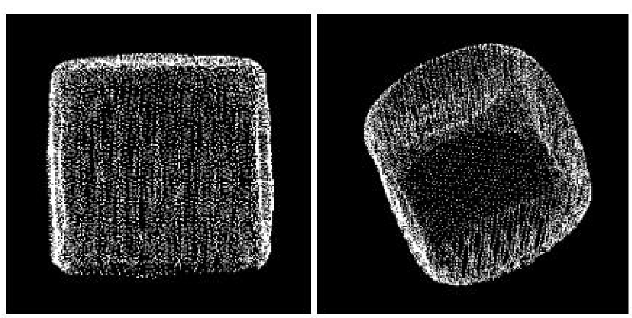


Figure 3.3: A cube with complete spatial information from its model

view of 360º (e.g., the cube toy in Figure 3.3). These models can be created by mathematical modeling or generated off-line by the VZX system. The locations and orientations of these models are fixed and known with respect to the robot base frame. Therefore, upon a successful matching, the relative pose of the specified object with respect to the robot frame can be obtained.

Certain properties of the objects, such as the preferred pick-up orientations in object coordinates, are also associated with their models. For a cube, as an example, the preferred pick-up orientations in its own frame can be 0º, ±90º, and ±180º. Therefore, once the orientation of the object is determined in the robot frame, the pick-up orientation for the robot gripper can be easily calculated. Multiple solutions may often exist, but only the most convenient one for robot gripper to reach is selected. Then based on robot inverse kinematics, the robot joint trajectories are calculated.

The orientation of a gripper when it approaches and is about to pick up an object is the same as the orientation of the object as defined above. Because the operations of fingers take some space, the surrounding of the object can be a factor to change the orientation of the gripper. An algorithm to check surrounding of a selected object will be developed providing factors to determine gripper orientation.

3.4. DETERMINING GRIPPER OPERATION

There are two basic operations for either a human hand or a robot gripper; they are "to open" or "to close". In robot path planning, these operations are determined based on the three primitive paths described in Section 3.2. An open-gripper command is issued before the pre-pick path to make sure the gripper is ready for pick, and a close-gripper command is given after the path being completed. Then before the home path, one more open command is provided to release the grasped object from the gripper.

For gripper closing operation, an appropriate grasping force has to be identified. This force is supplied by the REFES program with the command "C\n+020.000\n" (see Section 2.2.2), where the value of 20 is the specified force (~ 1.1 kg). Then in RobotManager, the grasping force is implemented in the robot language as "1;1;EXEC HCLOSE 1,63,20,.25" (see Section 2.7), where 63 is the initiating force (~ 3.5 kg) for a faster gripper operation and it lasts for only 0.25 seconds followed by the constant grasping force.

4. ARM MOVEMENT ASSISTED CONTROL

In this Section a number of tasks are identified in this section that improve the control of arm movement. They are:

1. Tracking the moving arm;
2. Identifying the possible obstacle and
3. Providing a dynamic trajectory for the controller.

To achieve these goals, fast image capture and processing are critically important. In order to track positions of a moving arm or a moving object, an image sequence that describes the object 3D status and their changes would be used. The accuracy of the motion tracking is based on a number of factors, including:

1. The accuracy of the feature identification and how fast the identification process can be achieved,
2. The effectiveness of the tracking algorithms and how expensive the computational time is,
3. And, even before all of the algorithms are developed, the speed that imaging system can process images.

Image capture speed in this phase was substantially increased.. Several design approaches were developed, tested and evaluated in this task.

4.1. EFFICIENT IMAGE CAPTURE

Image capture and processing speed is a key factor. How precisely the arm motion parameter can be generated depends on how effective a motion estimation algorithm works. This, in turn, depends on how well the algorithm is designed and how fast and how precisely the motion can be measured. How fast and how precisely the motion can be measured depends on how fast the cameras can produce the observation images and what resolution of the image the camera produce and how effective the image processing algorithm works. As the robot arm can move as fast as 1287 mm/second, 130 set of motion data in a second may be needed to achieve the fastest motion estimation if the sampling interval is 10 mm. This requires a camera speed of 130 frames/second and the capability of processing the same number of images each second. To limit or reduce the robot motion speed would help in solving the current slow image capture and processing problem. On the other hand, increasing the speed of the image capture and processing is a better solution. For instance, if we set the robot moving speed at 10% of its fastest speed, it is still necessary to have an image capture and processing speed of 13 images per second. Different approaches to increasing the image capture rate and processing were examined and incorporated where appropriate. Cameras with differing sensor resolutions were also used during testing.

4.2. HAND MOTION TRACKING

Hand motion tracking and motion parameter identification by 2-D camera projection was completed successfully. The hand motion was tracked by using both marker tracking methods and skin color and hand segments tracking methods. Both tracking and motion identification methods are verified by the demonstrations. The marker methods were tested and the tracking results analyzed by Ardem Medical, Inc.. (See Appendix VI, Testing Final Report)

4.2.1. BACKGROUND AND MOTIVATION

The arm motion controlled by FES system is noisy, slow and has a large amount of error. A position and orientation feedback compensation for the FES can provide a more precise control of the hand motion. Algorithms developed in this task are to:

1. Facilitate the use of cameras to track the current position and orientation of a moving hand
2. Predict position and orientation for the next step
3. Provide feedback to the hand motion control system
4. Minimize the deviation from the desired moving trajectory.

The algorithm used to update the prediction is based on minimizing the geometric error between the prediction and the observed past positions and orientations from the image frame sequence.

Hand motion tracking is a very important area in computer vision applications. The general solutions to this problem are divided into two categories based on the type of sensors used. The two categories are location sensor based hand tracking and vision sensor based hand tracking. In location sensor based hand tracking, sensors such as magnetic tracking equipment are used to identify the 3D hand position. In vision based hand tracking, sensors such as cameras are used to capture 2D views of a moving hand. The latter is widely applied and uses passive sensing, to capture natural motion.

Previous work on vision based hand tracking can be divided into 2D feature-based tracking and 3D model-based tracking. Methods in 2D feature-based register the possible 2D appearances of the moving hand target and find the best matching from the input images [12] and [13]. Methods in model-based tracking extract local image features and fit a given 3D shape of hand model to the features [14] [15] [16]. B. Stenger [17] used an Unscented Kalman filter to track the hand motion based on a 3D hand model. The model was built from quadrics that approximate the anatomy of a human hand. The filter is used to update its pose in order to minimize the geometric error between the model projection and a video sequence on the background. L. Tsap [18] used a 6 DOF model to simulate motion of a hand and used non-linear Finite Element Method (FEM) to analyze the hand motion from range image sequences. While model-based tracking provides details of finger motion, feature-based tracking deals primarily with hand position. In feature-based tracking, markers, grooves and skin color are used to identify the hand location and algorithms are developed to reconstruct the motion based on the changes of corresponding features over time. Wang [19] devised a set of techniques for segmenting images into coherently moving regions using affine motion analysis and clustering techniques. The scene is divided into four layers, and then the entire sequence is represented with a single image from each layer and with associated motion parameters.

It is assumed that the arm motion controlled by FES system is slow and the finger operations are as simple as open and close. This largely simplifies the hand-tracking problem. This enables us to focus on hand position and orientation tracking of a moving human hand regardless of the finger motions. At this point, feature-based methods are a good fit and therefore are used in this task. Among many available feature-based methods, marker and hand skill color are two of common feature-based methods can be used in this task. Practically, it would not cause too much trouble to wear a small marker on an arm and there is no additional requirement to track a hand skin as long as the hand is not pained by other color or wearing a grove. The advantages and disadvantages of using marker tracking compared with skill color matching are:

Methods:	Advantages:	Disadvantages:
Marker	Simple computation Precise Faster	Inconvenient Can't track hand orientation
Skin color	Can track hand orientation No marker needed	Complex computation Slower

The reason for using two methods in this task is to test the above advantages and to make it available in future tasks that may require the use of higher accuracy tracking of both hand position and orientations.

4.2.2. HAND MARKER AND HAND MODEL CONFIGURATION

In order to track the motion of the robot arm, a simulator of a hand, using marker-tracking methods, markers were designed and attached to both robots at SIS and CWRU. There are number of factors in the marker design that significantly affect the tracking accuracy. These include the size of the marker, what is the outlook of the marker, what color is the marker, and where the marker is attached.

4.2.2.1. MARKER COLOR

Every color consists of a number of different colored components. Some color components are very sensitive to lighting characteristics. The marker position changes when it moves while the robot arm or the hand is moving. The changes of the marker position lead to changes of the light the marker observes. The changes of the light observed can dismiss some color components and therefore, miss segment of the color region. This can largely reduce the hand marker or hand motion tracking accuracy. Different colors of the marker have been tested and red color is selected because its color components are easy to detect compared with other colors.

4.2.2.2. SIZE AND OUTLOOK OF THE MARKER

The size and outlook of the marker affect the outcome of marker region segmentation. See Figure 4.1. Therefore the selection of the size and outlook of the marker affects the accuracy of the marker motion tracking. The larger the size of the marker the robot arm or the hand has, the more accurate the marker segment can be extracted and the more precise the motion can be detected. On the other hand, the larger size of the marker the robot arm or the hand has, the smaller the moving space the robot has to operate.

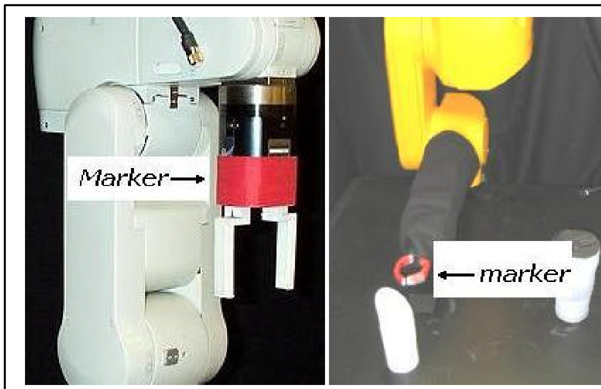


Figure 4.1: Red color markers are used to indicate the end-effector position of RV1A robot and to cover the fingers of RX60B robot.

The outlook of the marker can affect the position tracking as well. When the marker moves, the view from the tracking camera also changes. The outline of the marker can be changed with a projection operation from 3D space to 2D space. For position tracking with two cameras, simple geometric outline of a marker such as a sphere can be a perfect outline because the outline of a ball would not change its outline after a projection from 3D space to 2D space.

In this task, we designed a large size of the marker but not larger than the size when the end effector's finger is open. For the marker outlook, we used a ring on the RV1A robot and used the curve of the fingers on the Staubli robot RX60. The outlook of the markers is illustrated in Figure 4.1.

4.2.2.3. MARKER LOCATION

The locations selected for the markers on the robot arm and on the hand effect tracking results. The best position to attach the marker to is one that allows its observation during the entire tracking evolution. Consider that the tracking cameras are fixed and the workspace of robot arm operations is known. A location close to the finger or the hand might be a better choice. The marker was attached to the robot end-effector on the RV1A and simply used the finger as the marker in Staubli robot. The outlooks, colors and attached locations of both robots are shown in Figure 4.1.

4.2.2.4. HAND MODEL ATTACHED TO ROBOT END-EFFECTOR

The image processing procedures can be simple and fast if the marker is used to tack the motion. In this case, the outline and color of the marker can be tested and selected. However, when the hand moves, the marker can be covered by a part of the hand. During a normal operation of the REFES system, the hand can be obscured in several ways, e.g., by fixed objects during movement along a trajectory, by the user's own arm, or by objects moving into the scene. When the view of the marker is covered, the tracking will be interrupted or missing. The discontinuity of an error in of the tracking position information due to camera view blockage causes the robot simulator to move in an erratic and possibly dangerous manner. To avoid the missed tracking, a hand model is used to replace the marker and is attached to the end-effector. The offhand model attached to RV1A robot as shown in Figure 4.2. The marker needs to be attached to some part of the hand, and most likely, part of the hand behind the fingers. The marker within a camera view can be easily obscured by the fingers when the fingers are in front of a camera and the marker is on the other side of the hand. An additional tracking camera could be mounted at the top of the 3D workspace and facing down. But assuming a simple hand operation in this phase, the design of additional tracking camera is left as an option for the future development.

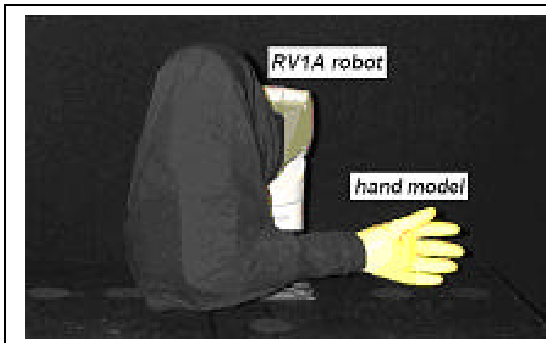


Figure 4.2: A hand model is attached to RV1A robot in SIS

4.2.3. HAND MARKER SEGMENTATION

The goal of image segmentation is to classify each pixel in an image into one of a discrete number of color classes called segments. Frequently the number of allowable segments is small; hence, the effect of segmentation is often the identification of the prominent features in an image. Several methods exist – including nearest neighbor and threshold. In the work described here, a pixel is thresholded according to its Red, Green, and Blue (RGB) components. The limits and threshold values are found by constructing histograms or RGB values for the entire image [23][24].

In this task, human skin color segmentation associated with dynamic shareholding method is used to track the hand region in a color image. The basic idea of current color segmentation techniques is to define a color class, and then use it for neighbor classification based on color space thresholding and probabilitistic methods. However, these techniques are susceptible to errors due to lighting. In particular, an incorrect color class may be identified or some critical part of the hand region may be missing.

If two images of the same scene are segmented according to the same threshold values, the various segments form a set of features that can be matched between the two images. For proper feature matching, it is important that the lighting in the images is as nearly the same as possible, and the cameras have similar color settings. Worse results are obtained if color interpretation differs between cameras. A more practical solution for this problem is to use a dynamic shareholding method. In this method, a factor is weighted to a related local color component of a pixel and used to pick a related component. The selection of a pixel is a local issue. Therefore, the lighting in the image can be varying to some degree. In this method, two color classes are used as a set of comparison. The following pseudo code illustrates the process:

$$if\ ((R >= Rlowerthresh)\ AND(R <= Rupperthresh))$$

$$AND(G >= a * R)\ AND(G <= b * R)$$

$$AND(B >= c * R)\ AND(B <= d * R)$$

$$pixel\_color = color\_class;$$

Where a, b, c, and d are the weighting factors. In this decision-making algorithm, instead of using fixed threshold values for G and B as in [23], relative comparisons are carried out between G/B and R, thus the impact of lighting conditions is significantly reduced.

$$if\ ((R >= Rlowerthresh)\ AND(R <= Rupperthresh))$$

$$AND(G >= Glowerthresh)\ AND(G <= Gupperthresh)$$

$$AND(B >= Blowerthresh)\ AND(B <= Bupperthresh)$$

$$pixel\_color = color\_class;$$

For more effective computational processing, selected color material was used to cover the background in this development. Figure 4.3 shows hand color segmentation from a pair of images

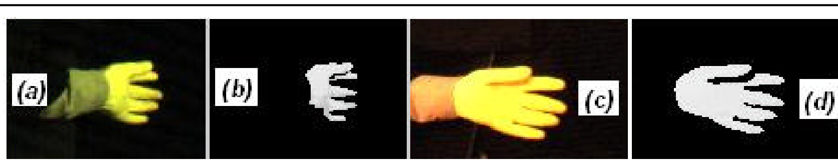


Figure 4.3: Hand region segmentation based on skin color using dynamic shareholding method. Image (a) and (c) are origin images. Image (a) is with respect to left camera coordinate and (c) to right camera coordinate. Images (b) and (d) are hand segments from (a) and (c).

taken from left camera and right camera. The background of the images is cover by black color material. The hand model was painted with yellow color and the color of the clothes to cover the forearm was selected different from the hand model color. The segmentation algorithms developed in this phase shows it distinguish the model color from others well.

4.2.4. HAND POSITION TRACKING

Hand motion tracking includes identification of moving hand positions and orientations over time. The process of identification is to reconstruct a sequence of 3D points from 2 sequences of images captured from the tracking cameras, the left and right cameras. There are two steps to achieve this goal: first the 2D positions are determined from the captured images, and second, the 2D positions from the two views are used to reconstruct the 3D positions.

The identification of hand moving orientation requires that we reconstruct a sequence of 3D points from the hand positions. A similar two-step procedure can do this, i.e., first, locate the changes of 2D position in current frame from the previous frame, and second, reconstruct the 3D position from 2D positions of two different views.

In order to locate the 2D position of a hand and to find the different 2D positions of the hand segments in the current frame from the previous frame, we need to find the hand segment in the image sequence. Hand region color segmentation is used in this task. Hand position and orientation tracking are presented in three sections:

1. Hand regions segmentation by using color image segmentation;
2. Position tracking; and
3. Hand orientation tracking.

Now we define the position of a hand to be the center of the hand in 3D space and assume that the projection of the hand position falls into the center of a hand segment. For hand position tracking using markers, an offset from the center of the marker to the hand center is calculated by measurement and was used to track the hand position. The 3D hand position with respect to the robot coordinates in the hand motion simulations system can then be calculated. Figure 4.4

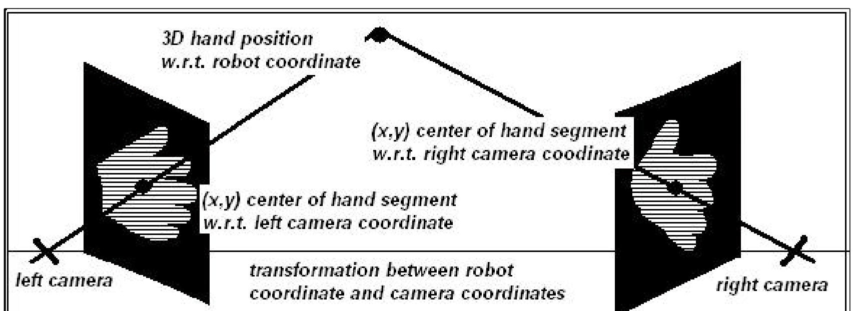


Figure 4.4: 3D hand position with respect to robot coordinate reconstruction from hand segments of left camera and right camera.

illustrates how the 3D hand position is constructed from two hand segments: The locations of two tracking cameras are determined during system calibration. Then the transformations from left and right camera coordinates to global coordinates (in this paper, global coordinates are referred to as robot coordinates) are used to transfer the center point of the hand segment from two tracking cameras to the robot coordinates. By finding the intersection of the two straight lines determined by the center point and the cameras, the 3D hand position is then calculated.

4.2.5. HAND ORIENTATION TRACKING

The goal of hand orientation extraction is to reconstruct the moving direction of the hand over time. There are no definitions of hand orientation used for hand tracking using markers. For hand tracking using skin color and hand model, we define the orientation of a hand by assigning the Z- axis to the middle finger direction and the Y-axis normal to and pointing away from the palm of the hand as shown in right picture of Figure 4.5. Hand orientation tracking is illustrated in the left hand part of

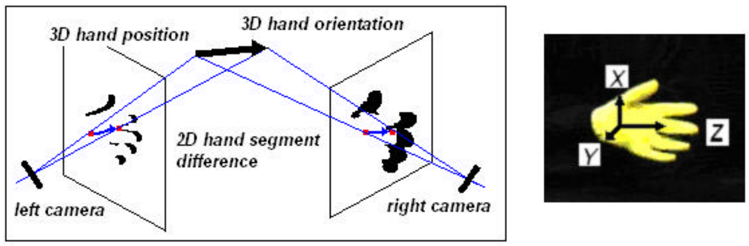


Figure 4.5: Hand orientation assignment and tracking from difference images between hand segments from frame to frame.

Figure. When a hand is moving over time, the difference between hand segments from frame to frame can be calculated. The regions of the difference indicate the 2D projection of a change in 3D space. It can be seen from the Figure that a hand has moved from left to right and rotated a positive angle about the Z-axis (assume that the Z-axis of the robot coordinate is up). This is evident from the motion vector. It can be seen from the figure that in this example, changes observed from the right camera is larger than the changes observed from the left camera. In this algorithm, the hand orientation tracking is calculated following steps:

a) At a time t, take images $I_{(L,t)}$ from left camera and $I_{(R,t)}$ from right camera.
b) Calculate hand segments $S_{(L,t)}$ from $I_{(L,t)}$ and $S_{(R,t)}$ from $R_{(R,t)}$.
c) Reconstruct hand 3D position P_t using methods discussed in Section4.2.4.
d) Calculate hand segment difference image $S_{(L,d)}$ by subtracting $S_{(L,t-1)}$ from $S_{(L,t)}$ and $S_{(R,d)}$ by subtracting $S_{(R,t-1)}$ from $S_{(R,t)}$.
e) Calculate 2D center points $C_L(x,y)$ from $S_{(L,d)}$ and $C_R(x,y)$ from $S_{(R,d)}$.
f) Reconstruct hand orientation 3D position Q_t from $C_L(x,y)$ and $C_R(x,y)$ using methods discussed above in this section.
g) The orientation of the hand orientation O_t at time t is the 3D vector connected P_t and Q_t.

4.2.6. HAND MOTION MODEL PARAMETER ESTIMATION

In order to recover the motion of a hand, it is necessary to construct a hand motion dynamic mode. The hand motion dynamic mode is a mathematical model describing the changes of the hand position state from time to time. How accurate the hand motion dynamic mode can describe the motion of a hand is based on two factors: how suitable the dynamic characteristic mode is created and how precisely the parameters of the mode are selected. The discussion in creation of hand motion dynamic mode and selection of mode parameters is divided into two sections: 1) hand motion model and its parameters; 2) hand motion model parameter estimation.

4.2.6.1. HAND MOTION MODEL AND ITS PARAMETERS

If we focus at its positions rather than its gestures, the hand motion can be described by the motion of a point in 3D space The motion of a 3D point can be represented as changes of state of a 3D point. The changes of a state can be described by a dynamic system. As a dynamic system, the motion can be described by the evolution of a set of state variables that consists of three rotation parameters and three translation parameters. Let $X = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}'$ be a point on the curve that describes the motion of a point. Subsequent points on the curve may be obtained by rotating X by a small angle a about some axis, says A, and then by translating the rotated point by the translation vector $T = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}'$. The three translation parameters referred to above are t_x, t_y, and t_z.

To determine the three rotation parameters, the Euler angles, a_x, a_y, and a_z, are calculated for the axis A. Rotation about A by a is then equivalent to rotating about the Z-axis by a_z, then rotating about the Y-axis by a_y, and then rotating about the X-axis by a_x. The rotation matrix about the axis A is the product of the 3 rotation matrices about the coordinate axes. Hence the model for rotation followed by translation can be written as:

$$X_{t+1} = RX_t + T \qquad (4.1)$$

Where

$$R = \begin{bmatrix} v_0 & v_1 & v_2 \\ -v_1 & v_0 & v_3 \\ -v_2 & -v_3 & v_0 \end{bmatrix} \qquad (4.2)$$

The values v_1, v_2 and v_3 from the rotation matrix R are the three rotation parameters described above and v_0 is assumed to be 0 - along with t_x, t_y, and t_z form the set of six parameters [20].

To model the motion of a hand with noise, we add a noise vector to equation (4.1). Define the rotation and translation noise vector $e = \begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix}$, where e_1, e_2, e_3 are independent normally distributed random noise. Since the noise is additive, a transformation matrix c with diagonal entries c_1, c_2 and c_3 may be defined. Then equation (1) becomes:

$$X_{t+1} = RX_t + T + ce \qquad (4.3)$$

$$c = \begin{bmatrix} c_1 & 0 & 0 \\ 0 & c_2 & 0 \\ 0 & 0 & c_3 \end{bmatrix} \qquad (4.4)$$

4.2.6.2. HAND MOTION PARAMETER ESTIMATION

In section 4.2.6.1, a dynamic mode of hand motion was developed and a discrete state equation was obtained as shown in Equation (4.3). In this section we discuss how the parameters of the state equation, the rotation matrix R and translation vector T can be determined.

If we rewrite Equation (4.3) so that the parameters from rotation matrix R and translation vector T are separated from the state parameters which can be observed as previous discussion in section 4.2.6.1Obviously, Equation (4.3) can be written in the same format as a linear regression model of

$$X_{k+1} = f'_k q \qquad (4.5)$$

Where

$$X = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \qquad (4.6)$$

$$f' = \begin{bmatrix} x_1 & x_2 & x_3 & 0 & 1 & 0 & 0 & e_1 & 0 & 0 \\ x_2 & -x_1 & 0 & x_3 & 0 & 1 & 0 & 0 & e_2 & 0 \\ x_3 & 0 & -x_1 & -x_2 & 0 & 0 & 1 & 0 & 0 & e_3 \end{bmatrix} \qquad (4.7)$$

$$q' = \begin{bmatrix} 1 & v_1 & v_2 & v_3 & t_x & t_y & t_z & c_1 & c_2 & c_3 \end{bmatrix} \qquad (4.8)$$

Equation (4.7) is a parameter vector included six transformation and three noise parameters. The problem of transformation parameters estimation now becomes a problem of constructing a general mapping from a sequence of observation data, starting from the observed data, recursive methods pose further constraints on the computation of parameters estimates. It is easy to consider that by taking a guess and then modifying it. Thus, equations (4.5) can be easily solved by applying a recursive least squares algorithm [21] [22]:

$$\hat{q}_t = \hat{q}_{t-1} + K_t [y_t - f_t \hat{q}_{t-1}] \qquad (4.9)$$

$$P_t = P_{t-1} + P_{t-1} f_t [I + f'_t P_{t-1} f_t]^{-1} f'_{t-1} P_{t-2} \qquad (4.10)$$

Where

$$K_t = P_{t-1} f_t [I + f'_t P_{t-1} f_t]^{-1} \qquad (4.11)$$

With initial condition of P_0 positive $\hat{q}_0$ are given.

4.3. OBSTACLE IDENTIFICATION

Obstacle identification algorithms have been designed and developed based on information about the moving object identification and information from robot working environment manager and trajectory planning. During an operation, there are two type of objects appeared in the workspace: static objects and moving objects. The static objects can be operating targets or objects existing since the system started so that the system has knowledge of their existing. Besides the static existing objects, there could occasionally be some objects moving into or out of the robot-working environment – e.g. another person places a cup of coffee on the tray. If this happens before any movement of the robot arm, a re-capture of the objects can be initiated by the user with the VZX system as described in the previous section, thus no confusion or collision would occur in the path planning. However, if this happens when the robot arm is moving, the emerging object may become an obstacle and cause collision. To address this problem, an on-line monitoring system is implemented by using two separate cameras, which overlook the entire robot workspace, as illustrated in Fig. 4.6.

4.4. OBSTACLE AVOIDANCE

Obstacle Avoidance monitoring system serves a two-fold purpose: 1) to detect if there are any new or missing objects in the robot workspace when the robot was not moving and further to notify the user for necessary re-capture, and, 2) to detect any moving or emerging obstacles in the robot's planned path once a movement has commenced. Note that for the second function, only a fraction of the image of each camera is processed for obstacle detection. : 1) to detect if there are any new or missing objects in the robot workspace when the robot was not moving and further to notify the user for necessary re-capture, and, 2) to detect any moving or emerging obstacles in the robot's planned path once a movement has commenced. Note that for the second function, only a fraction of the image of each camera is processed for obstacle detection. This is done by aiming for increased processing speed for real-time reaction of the robot arm. The monitored area is focused on the robot-planned path just before the arm is scheduled to arrive. Figure 4.6 shows the concept. When an emerging obstacle is

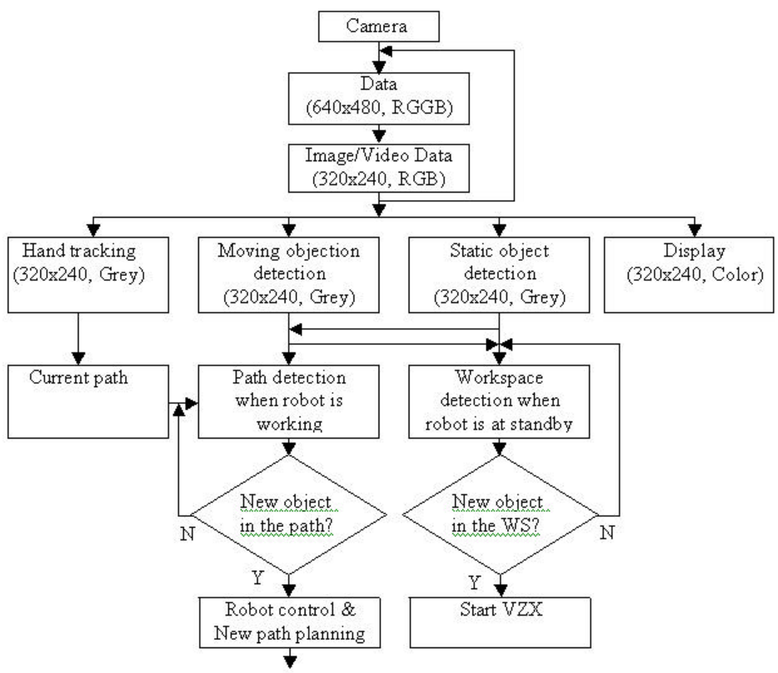


Figure 4.6: Obstacle identification and avoidance logical diagram.

detected, its size and location have to be calculated in order to re-generate a new collision-free path. Based on the image segments of the obstacle on two cameras, a stereovision algorithm is applied for this calculation.

Still, one more issue has to be addressed, that is the robot arm tracking. This is not only to provide feedbacks for FES closed-loop control, but also to tell where to focus along the robot path for obstacle detection. By using the ideas of stereovision and hand color recognition, the position and orientation of the robot gripper can be determined in real time when the robot is moving.

5. USER INTERFACE

The user interface was designed not only to allow easy specification of common tasks and presentation of the system state in a simple manner, also to allow more detailed information for a more knowledgeable user. The system was constructed in an manner that allows user-friendly interaction and interaction in a way that is straightforward for patients and end-users. Studies of user-accessible feedback of planning functions were undertaken and user-friendly task specification was provided. REFES interface with head tracker and voice was performed successfully in CWRU. Detailed discussion of user and patient tasks specifications, available interface methods and user interface hardware development can be found in Appendix V. A primary consideration for requirements of user interface of the REFES and neuroprosthesis system was ease of use for multiple models. This means either a head tracker or a voice recognition system can be added to the user interface. The agreement on the following interface specifications were reached after discussion with the staff of FES Center:

1. A graphic user interface (GUI) will be designed and developed and will be displayed whenever the REFES and neuroprosthesis system starts and will be displayed on the desktop while the system is running.
2. The GUI will provide the user of the REFES /neuroprosthesis system with a top (overhead) view of the robot workspace in front of them. In other words, a 2D graphical display of the robot workspace will be shown within the GUI. The 2D graphic of the robot workspace is the projection of robot workspace onto the robot-working table. With respect to the robot system coordinate, the display area is the robot workspace projection from the Z-axis to the X-Y plan. Objects within the scene will be projected and identified in their projected locations. These objects are to be listed in an on-screen menu or each object could be labeled at its location on the screen.
   The cursor is then placed over the label and the object selected using the mouse emulator.
3. The user will specify only the desired endpoint of a movement by selecting a location from the 2D graphics of the robot workspace. The REFES will compute the corresponding 3D location and send the location to the robot controller.
4. A number of buttons will be designed to facilitate the basic operations.

5.1. GUI DESIGN AND SOFTWARE DEVELOPMENT

Two different versions of GUI were designed and developed for REFES system in SIS and FES Center in CWRU. The difference between the two is the 2D graphic robot workspace due to differences in the two robot systems. Figure 5.1 shows the GUI designed for REFES system at SIS and Figure 5.2 illustrates the GUI designed for robot system installed at the FES Center in CWRU. From both figures, we can see a number of selection buttons designed for the GUI. A "Move To Position" selection button at the top of the window is designed to allow the

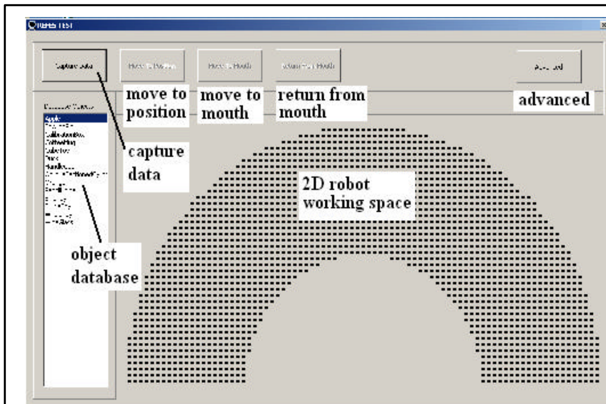


Figure 5.1: GUI designed for SIS REFES.

selection of motions for the robot manipulator to complete. A "Capture Data" button in the upper left of the main window is designed to capture the image data to determine the object's position within the workspace. An "Advanced" button in the upper right corner of the GUI is designed to activate this setup process. A list of objects within the "Object Database" on the left of the window is designed to display all types of object in the database and allow the user to select an object to be operated. A "move to mouth" selection button at the top of the window is designed to allow the robot arm to move an object to a mouth position that is given at the beginning of the operation and a "move from mouth" selection button on the top of the window is designed to allow the robot arm to move the object away from the mouth position. Details descriptions of the GUI commands and operations can be found in Appendix II**.**

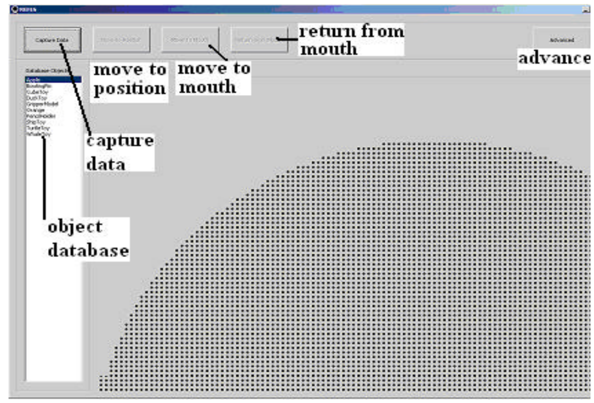


Figure 5.2: GUI designed for REFES at FES Center at CWRU.

5.2. USER INPUT INTERFACE TO ROBOTEYES™ SYSTEM

As described above, the selected head tracker mouse emulation systems interface directly with standard computer mouse ports, so no hardware development was necessary. The QPointer voice recognition software is an add-on to Windows, but it requires no special software development because uses the built-in voice recognition features of the operating system. QPointer works by identifying text tags of features on the desktop and any open windows. Various types of text can be displayed, including icon labels, button labels, and window names. Recognized text is indicated by a pop-up number next to that text. Multiple instances of the same word are sequentially numbered, starting at zero. Speaking the number of the text you wish to select places the cursor at that point and left-clicks on the text. RobotEyes™ uses this feature by having all items in the interface identified with unique text tags. QPointer is used to activates the various REFES functions simply be speaking the name of the button or object to be manipulated and then its number.

6. **SYSTEM REQUIREMENTS**

In this section, the identification of FES range of motion, motion command parameters, and FES simulation construction are discussed and developed.

Compared with the robot manipulator, a REFES / neuroprosthesis system will provide a much smaller motion range, a much slower motion and a much noisier motion. Undoubtedly, the range of motion that the ultimate REFES / neuroprosthesis system will achieve will be limited primarily by the FES system, not by the REFES system for the following reasons:

1. The range of possible arm movements to shoulder level and below will be very limited because of muscle weakness due to disuse atrophy and denervation.
2. The radius of possible motions will be limited to the length of the user's arm because these individuals will also lack voluntary control of muscles of the torso.
3. Individuals with high tetraplegia will almost always perform functional tasks from the base of a lapboard or a tabletop.

Therefore, the likely range of motion that we can restore will be from approximately the lower abdomen to shoulder height, with a reach limited to the length of the arm. The functional tasks targeted by our neuroprosthesis reflect this expected workspace and are focused on restoring the ability to bring the hand to the mouth to allow feeding and grooming activities and to allow the arm to reach out to manipulate objects with the hand within the limited workspace in front of the body (e.g., to acquire food). The workspace provided by the Staubli robot used in this study closely replicates this limited workspace.

The maximum resultant velocity of the RV1A robot manipulator is 2200 mm/second with an average of angular moving speed of 159 degree/second while individuals with high tetraplegia will almost always perform moving tasks as slow as ten's mm/second and ten's degree /second. The pose repeatability and distance accuracy of the RV1A robot arm is between –0.02 mm to +0.02 mm while a human arm can hardly reach a distance accuracy of less than 1 mm. Larger errors than a normal human mar motion the neuroprosthesis system can have due to external disturbances, fatigue, or other changes in the properties of the system. In high tetraplegia, the entire upper extremity is paralyzed, so voluntary correction for errors in the performance of the FES system will be very limited (i.e., perhaps just shoulder shrug).

This will greatly facilitate the introduction of the REFES system into real neuroprosthesis testing when implanted human subjects are available in 12-18 months. Note that the robot used at CWRU is significantly different from the one in use at SIS, demonstrating the flexibility of the REFES system

Functional use of the REFES /neuroprosthesis system was demonstrated by a human user commanding the robot to perform three different tasks. The system was found to be straightforward to use and the movement times for the various tasks were well within functionally tolerable ranges.

With the completed work in this phase, the REFES system is able to account for the degrees of freedom and range of motion supplied by the FES arm and generates appropriate trajectories to perform user-specified tasks. While a lower speed and a smaller range of motion would not affect the control of the REFES system, a noisy motion from the neuroprosthesis system will certainly cost the successful operation. The dynamic characteristic parameters of the FES will be estimated and used to improve the motion control. Discussion of identification of FES range of motion and command parameters, construction of FES simulation and translation driver can be found in Appendix V.

7. DEMONSTRATION TEST 1: QUANTIFICATION OF DESKTOP RECOGNITION, PLANNING AND ARM LOCATION

This demonstration was designed to testing was to the ability of the REFES system to measure arm posture and desktop objects in progressively realistic environments. The demonstrations were performed both at the FES Center at CWRU and at SIS. They were also tested by Ardiem Medical, Inc. The anthropomorphic robot manipulator Staubli RX60B robot and Mitsubishi RA1A robot were used as simulated human arms moving within the robot workspace. The REFES systems were set up facing robot-workspace to observe the motion of the robot and desktop objects. The locations of the object were measured and compared with those identified by the REFES system. The accuracy of observation by REFES system was calculated after a series of repeating tests. The demonstrations have shown that the REFES system was able to identify 3D object location within the robot's workspace and was also able locate a selected object to a given location with a high accuracy, thus closely replicating the operation performed by an anthropomorphic system.

7.1. SYSTEM DEMONSTRATION AT SIS

The purpose of this demonstration and testing was to confirm the accuracy of the REFES in identifying the 3D objects within the robot'sw workspace. By using a cylindrical object, the tests challenged the ability of the REFES system to accurately measure the three dimensional location and orientation of the test object and to accurately transform that data to coordinates usable by a robotic arm.

7.1.1. DEMONSTRATION CONFIGURATION

The system configuration necessary to meet the requirements of the REFES system is the robot-working table, measuring approximately three feet wide by two feet deep, and the Mitsubishi RV1A robot mounted at the rear center of the table. This is robot arm is shown in the center of the photographic shown in Figure 7.1. The VZX system was mounted on the extension of the robot-working table and facing down to the robot-working table. The background, tabletop, and the robot arm with the exception of the gripper are black or draped in black. Two stationary cameras are mounted one each at the two front corners of the table. The stationary cameras are mounted approximately one foot from the table and approximately six inches above the table and are angled roughly toward the center of the robot base. The VZX camera is mounted directly above the stationary camera on the right side looking toward the workstation. Above the movable camera is a strobe illumination source that projects a pattern of vertical stripers on the robot and any 3D objects within the robot workspace. A monitor, keyboard, mouse, and computer are located on a table to the right of the RobotEyes™ system.

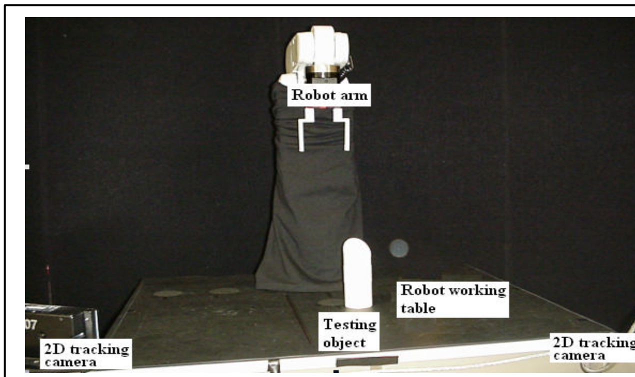


Figure 7.1: REFES configuration with Mitsubishi robot arm for desktop recognition demonstration and test in SIS.

7.1.2. DEMONSTRATION METHODS

The robot-workspace on the tabletop is a semicircular band approximately 180 degrees around the robot zero center point and centered on the zero point of the x axis as shown in Figure 7.2. The inside radius of the band is approximately 204 millimeters and the outside radius is approximately 417 millimeters as measured from the (0,0) point of the robot's base. Six of the ten zones used in the initial accuracy test were selected as locations for the test object. Three of the locations corresponded to the three highest error zones in the initial accuracy test and three of the locations corresponded to the three lowest error zones in the initial accuracy test as shown in Figure 7.2.

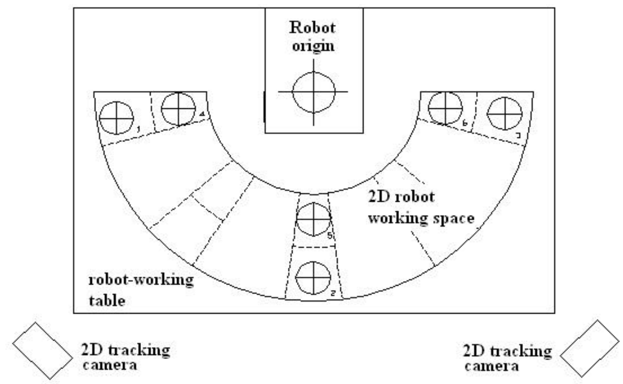


Figure 7.2: Locations and orientations testing with respect to the 2D robot-workspace.

The Mitsubishi RV1A robot is provided with factory machined surfaces on the robot base for x and y reference measurements. Raw x and y coordinate measurements were taken from these reference planes on the base of the robot. From the reference planes the center of the J1 axis can be accurately determined. The robot coordinate system uses the center of the J1 axis as the origin for x and y coordinates. The z-axis was constant for the object and was measured as zero at the plane of the work surface. A cylindrical object with an oblique angled top was used in each test and the x and y-axis were measured to the centerline of the object.

The coordinate of the REFES system is defined its Z-axis pointing to the focus of the camera and its Y-axis pointing to the top. Its Z-axis is defined with respect to the Y-Z plane based on the left hand rule. During the initial scan of the workspace the robot gripper was used as a calibration object for the system. All REFES coordinate references are relative to this object location and orientation. The REFES coordinates were invisible to the test coordinates. All coordinates measured and calculated are referenced to the robot coordinates.

During the test, six black disks with orientation lines were accurately placed on the work surface with the orientation lines parallel to the X and Y-axes. The disks were held in place with double-sided tape. A cylindrical object 58 mm diameter and 160 mm height was placed on one of the target locations and centered (see Figure 2.8 in Section 2.2.4 for drawing of the test object). The object was sequentially oriented about the center of the disc in five angular orientations. Zero degrees, plus ninety degrees, minus ninety degrees, plus forty-five degrees, and minus forty-five degrees. Zero degree orientation was set to be with the center of the plane of the oblique angle facing away from the robot and perpendicular to the y-axis of the robot. After each orientation of the robot at the object location the RobotEyes™ system was activated to collect data on the location and orientation of the object. This was repeated for each of the five angular orientations. The process of orienting and using the RobotEyes™ system to obtain data were performed at each of the six locations. The series of five orientations at the six locations was repeated a total of three times at each location yielding a total of three sets of five orientations for each location.

7.1.3. DEMONSTRATION RESULTS

Table 7.1 summaries the average composite error of all 90 designed locations during the tests. Consider the operation error in the Z-axis can be neglected because of the gripper operations of both robots, the average position error is 8.68 mm which is smaller than the requirement of 10 mm while number of test with an error value of larger than 10 mm. The rotational error is smaller than 1.5 degrees. For the current operational requirement, Simple object such as the coffee mug doesn't require a specified operation orientation. For pick up operations, an error of 1.5 degree is reasonably small and would not affect the pick up operation.

Table 7.1: Overall Average Composite Error for All Locations

	X	Y	Z	Yaw
	mm	mm	mm	deg
Average Error	-8.68	0.52	-3.31	-1.44
Standard Deviation	4.03	3.64	1.63	2.64
Minimum Value	-15.3	-4.7	-8.6	-8.9
Maximum Value	-3.0	6.4	-0.4	4.7

7.1.4. ERRORS

The following sections are discussion of the errors that were observed during the testing.

7.1.4.1. LOCATION ERROR

The data seems to indicate that the x and y-axis errors are very consistent at a single location but that the average error may vary between locations. The x-axis exhibited the greatest variation between locations. The least error in the x-axis was recorded at locations one and four that correspond to the two locations furthest from the coordinate camera. Locations one and four are to the left of the robot as observed from the perspective of the coordinate camera. The greatest error in the x-axis (greater than 10mm) was recorded at locations two, three, and six that correspond to the locations closest to the coordinate camera. Location five exhibited an x-axis error value that was roughly between the highest and lowest error value locations. Interestingly all of the x-axis errors are of a negative value regardless of location. The y-axis error was very low with the overall average error less than 1mm at a 3.64 mm standard deviation. A location specific trend pattern similar to the x-axis was noticed in the y-axis. Locations one and four exhibited positive location errors and locations two, three, five, and six exhibited negative location errors.

The location errors in the x-axis exhibit the greatest error level (greater than 5 mm) in all except two locations. The y-axis error generally falls within a +/-5mm error range regardless of location. Minimally an adjustment to the transformation algorithm to compensate for the x-axis error would be suggested. From a practical standpoint the use of the RobotEyes™ system with a FES equipped patient at the accuracy level determined in this test should be more than adequate. For greater accuracy and use with a robot the x-axis errors should be corrected to within a +/- 5mm range.

7.1.4.2. ORIENTATION ERROR

The orientation (yaw) errors appear to be relatively small and somewhat random in nature. A small but noticeable trend was observed in the data. At some locations the negative rotations specifically minus ninety and minus forty-five degree exhibited a slight but noticeable increase in rotational error. The locations that exhibit this trend appear to correspond to the furthest and closest locations to the camera. While this trend is noticeable from the data the practical result is that the yaw error is of a sufficiently low order that the effect of the error should be minor.

7.1.4.3. OTHER ERRORS

Error caused by roll, pitch, and the z-axis generally was a low error value. The degree of variation between zones and in individual data points was very consistent and was of a sufficiently low order that the effect of the error should be minor.

The testing process was interrupted by several random data collection defects that were attributable to computer hardware. The data files had good data except for one random disturbed frame that rendered the information unusable. Data collection had to be repeated for the failed files. A new computer would be advisable for future tests to reduce or eliminate this reliability issue.

7.2. REFES DEMONSTRATION IN THE FES CENTER OF CWRU

The second demonstration and test was performed in the FES Center at CWRU. is to test the ability of the REFES system to accurately measure the 3D location and orientation of an object within the robot-workspace and to accurately translate that information from the camera coordinate system to the robot coordinate system such that the objects can be operated by the robot arm.

7.2.1. DEMONSTRATION SYSTEM CONFIGURATION

The system was set up based on the configuration requirements from the REFES system. A robot-working table was set up and the Staubli robot mounted suspended above the rear center of a table measuring approximately 991 mm by 813 mm depth. This is shown in Figure 7.3. The VZX system was mounted on the extension of the robot-working table and faced down to the robot-working table. The background, tabletop, and robot arm., with the exception of the gripper, were black or draped in black. The robot gripper is a scissors type of claw with the opposing legs of the claw curved towards each other to facilitate the grasping a cylindrical object. Across the table and facing the robot arm, two stationary 2D tracking cameras detecting the motions are mounted, one each at the two front corners of the table. One of the 2D tracking cameras was mounted approximately one foot or less away from the table, approximately six inches above the table, and angled roughly toward the center of the robot claw. The other was mounted just below the front of the VZX. Both 2D tracking cameras are focused at the center of the robot-working table. A stripe projector source was mounted directly above the stationary camera on the right side looking toward the robot claw. A 2D projection of the robot-workspace to the working table and test object locations are shown in Figure 7.4.

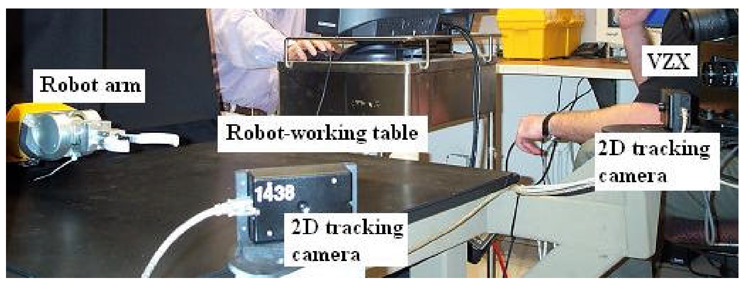


Figure 7.3: System setup with Staubli robot arm for desktop recognition demonstration and test in the FES Center at CWRU.

7.2.2. TEST METHODS

A flat black matte board with location target circles drawn on the board to locate and oriented the cylindrical test object was affixed to the worktable surface with double sided tape (see figure 7.4).

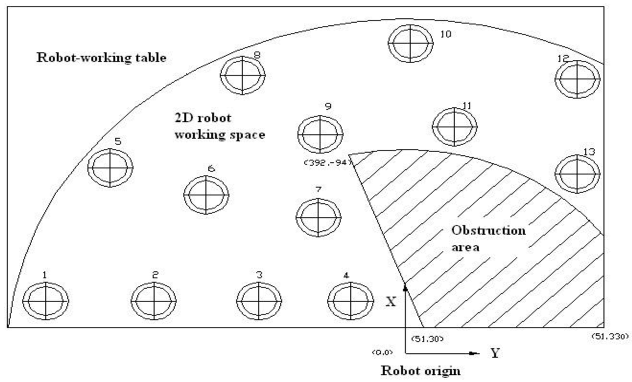


Figure 7.4: Locations and orientations testing with respect to the 2D robot working space in the FES Center at CWRU.

The orientation lines on the target circles were oriented to correspond to the x and y-axis of the robot. Locations one and thirteen were outside of the view of the REFES system and were not used in the test. During the test, A cylindrical object 58 mm diameter and 160 mm height was placed on one of the target locations and centered. The test object was sequentially oriented about the center of the target location in five angular orientations. A number of degree placements set to 0, 90, -90, 45, -45 was used repeatedly. Orientation of 0 degree was set to be with the center of the plane of the oblique angle facing away from the robot and perpendicular to the Y-axis of the robot. After each orientation of the object the REFES system was activated to collect data on the location and orientation of the object. This was repeated for each of the 5 angular orientations. The process of orienting and using the REFES system to obtain data were performed at each of the 11 locations. The series of 5 orientations at the 6 locations was repeated a total of 3 times at each location yielding a total of 3 sets of 5 orientations for each location. A total of 165 tests were performed and the results were recorded and calculated in next sub-second. The physical coordinates of the target locations were recorded manually on a data chart and the RobotEyes™ data was electronically collected and stored in a separate individual file for later transformation to robot coordinates.

7.2.3. TEST RESULTS

The test results of total 165 iterated tests are calculated as an overall average composite of all locations. Coordinate data from the physical dimensions were compared with the output data generated from the REFES system. The differential between the actual measured locations and orientations and the robot coordinates and orientation was then calculated. For the location differences, the position of measurement is defined as the center of the bottom circle of the test model cylinder with respect to the robot coordinate. The model position of the estimated by the REFES system is the center of the 3D point model on the robot working-table with respect to the robot coordinate.

Table 7.3 Overall Average Composite of All Locations Less Anomalous Readings

	X	Y	Z	Yaw
	mm	**mm**	**mm**	**deg**
Average Error	11.33	-12.65	-0.98	-1.98
Standard Deviation	25.25	14.65	2.02	1.12
Minimum Value	-32.08	-47.94	-4.95	-4.00
Maximum Value	65.68	14.22	3.01	-0.06

While the average error in orientation identification is small, the data reveals that the position identification error values with the Staubli robot were considerably large but still, the error is under the error limit of an operating condition. On the other hand, the average error values with the Staubli robot were considerably larger than the error values obtained with the Mitsubishi robot (see section 7.1.4). Few of the locations exhibit error values greater than the operating limit of absolute value of 20 mm.

The data seems to indicate that the location errors (x and y-axis) are partly caused by distance from the centerline of the REFES camera. The locations that exhibit the highest accuracy are those closest to the centerline of the REFES camera. Some distortion apparently occurs at the extremes of the camera view.

Errors attributable to the end-effector used on the Staubli robot were also suspected. The end-effector was constructed and oriented differently than the end effector that was used on the Mitsubishi robot and there was more difficulty in obtaining accurate coordinates of the gripper.
Some locations exhibited a slight correlation between error values and object orientation. This could have been a random event or could be evidence of the coordinate transformation matrix requiring some modification.

Three random occurrences of anomalous data for three separate locations could not be adequately explained. The anomalous data was notable for being grossly inconsistent with other data in the series. The data anomaly could be software or hardware related or even human error.

7.3. DEMONSTRATION 1 DISCUSSION

The demonstrations and tests results indicate that the REFES system is capable of recognizing 3D objects within a robot workspace. It shows a great accuracy in 3D object information identification. Under the requirements of robot operation within its workspace, the identification accuracy satisfies a safe robot operation. The test results also show that the identification accuracy in system set up and working environment in SIS is considerably higher than what had been obtained by tests as evidenced by the results from the Staubli robot in FES Center at CWRU. The location of the object in relation to the REFES camera were improved primarily by modification of the transformation matrix. The end-effector is the most prominent difference between the two robots and may be main cause of the high error values observed.

8. DEMONISTRATION TEST 2: USER INTERFACE AND FEEDBACK-ASSIST ARM POSITION CONTROL

The purpose of this demonstration was to test how the user interface works with the system and how the system provides feedback information during its applications. The user interface test was performed by the REFES system with the Staubli RX60B robot arm in CWRU. A voice recognition system and a head tracker mouse emulator between the human user and the REFES system were used to test the system. The details demonstration test results and discussion can be found in Appendix V of this document. The demonstration test of the feedback assist arm position control was performed on the REFES system with Mitsubishi RV1A robot arm at SIS in January 29, 2004. The arm motion tracking - using both methods with and without markers - were demonstrated. The test and demonstration results indicate that the tracking accuracy satisfies the position feedback control. While the average of tracking error is no larger than 8 mm over a consistent and repeatable error value curve, few errors can be larger than 25mm of tracking limitation. The image captured speed for matching is larger than 20 frames per second. This speed satisfies the matching requirement of a sufficient rate for neuroprosthesis applications.

8.1. ARM POSITION FEEDBACK CONTROL DEMONSTRATE

The purpose of this demonstration was to test the ability of the REFES system to act as a feedback control for FES by tracking the robot arm and test object as it is being moved between locations within the robot workspace.

8.1.1. DEMONSTRATION CONFIGURATION

The system was set up based on the REFES system configuration requirements and is illustrated in Figure 8.1. A Mitsubishi RV1A robot was mounted at the rear center of the robot working-table. The background, tabletop, and the robot arm with the exception of the gripper are black or draped in black. A red band of tape is wrapped around the diameter of the end-effector to facilitate tracking of the robot end point location just above the robot gripper. Two stationary 2D tracking cameras (motion detection cameras) are mounted one each at the two front corners of the table. The cameras are mounted approximately 1 foot or less away from the table and approximately 6 inches above the table and are angled roughly toward the center of the robot base. The VZX imaging system of the REFES system is mounted directly above one of two stationary 2D tracking cameras on the right side looking toward the middle of the robot working-table. Above the VZX imaging system is a stripe projection source. A computer monitor, keyboard, mouse, and computer are located on a table to the right of the workstation and are used to operate the RobotEyes™ system.

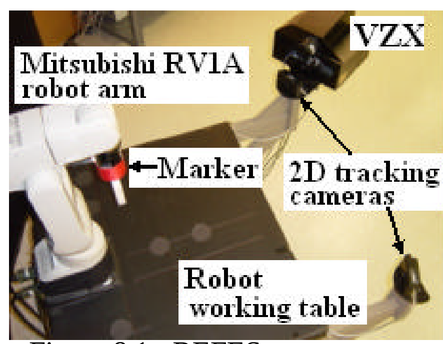


Figure 8.1: REFES system set up with Mitsubishi RV1A robot arm for arm tracking demonstration and test.

The robot workspace on the tabletop is a semicircular band approximately 180 degrees around the origin of the robot coordinate and centered on the origin of the X-axis. The inside radius of the band is approximately 204 millimeters and the outside radius is approximately 417 millimeters as measured from the 0,0 point of the robot.

8.1.2. DEMONSTRATION METHODS

For the demonstration and test, test objects were placed on the robot-working table. User input was given to the system from the graphic user interface to move an object to a destination location, with or without obstacle. The REFES system picked up the object and moved it to the desired location and outputted two sets of trajectory points: the robot gripper moving positions and the tracking positions during the operations. The reason for using robot gripper positions from the REFES system is the difficulty in measuring a moving position. Figure 8.2 illustrates examples of plot out of the robot gripper moving position compared with the tracking. One of the test patterns of the robot arm tracking motion designs is to select an object and give a terminal position passing over a static obstacle. The REFES system controlled the gripper started from 1) the gripper home position, and moved down to the 2) object position and then, grasped the object, and moved over the 3) obstacle position, and moved to 4) the terminal position, then opened the gripper and released the object, and returned back to 1) gripper home position. The moving positions of the gripper were plotted as the red dots within the robot workspace as shown in the Figure. The tests were repeated 10 times and the details of the records can be found in Appendix VI of this document.

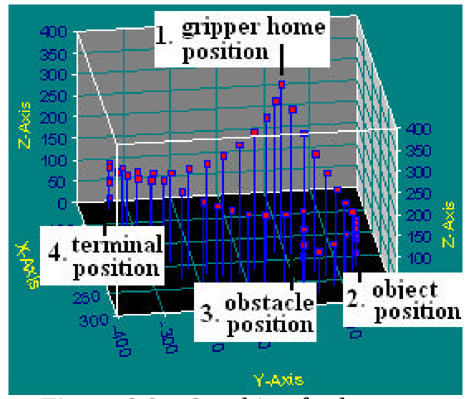


Figure 8.2: Graphic of robot arm tracking motion design.

8.1.3. DEMONSTRATION RESULTS

Table 8.1 summaries the average composite error of all 10 designed repeatable tracking during the tests. The average position tracking error is about 11 mm that is slightly larger than the requirement of 10 mm while number of test with an error value of larger than 25 mm.

Table 8.1 Overall Average Composite Error of 10 tracking tests

	X-Axis Robot - Tracking Differential	Y-Axis Robot - Tracking Differential	Z-Axis Robot - Tracking Differential
Average Error	11.05	-11.716	0.288
Standard Deviation	9.108	6.778	7.63
Minimum Value	-4.943	-24.079	-14.173
Maximum Value	23.761	1.321	17.992

One of the tracking test results is plotted and illustrated in Figure 8.2. Plot (a) shows the plots of comparison between the measurement positions and the tracking positions with respect to the Y axis as a function of time. The motion was limitted to a sub-space between about –250 and +250. While the tracking positions are close to the measurement positions between athe time of 35 and 85, there is a larger error exists between time 10 to 30. Figure 8.2. Plot(b) shows the comparison between the measurement positions and the tracking positions with respect to the X axis as a function of time. The motion was limitted to a sub-space between about 20 and +250. While the tracking positions are close to the measurement positions between athe time of 30 and 60, there is a larger error exists between time 15 to 30 and from 55 to 70. Figure 8.2. Plot (c) shows the plots of comparison between the measurement positions and the tracking positions with respect to the Z axis as a function of time. The motion was limitted to a sub-space between about 480 and +760. The tracking positions are close to the measurement positions during the test time.

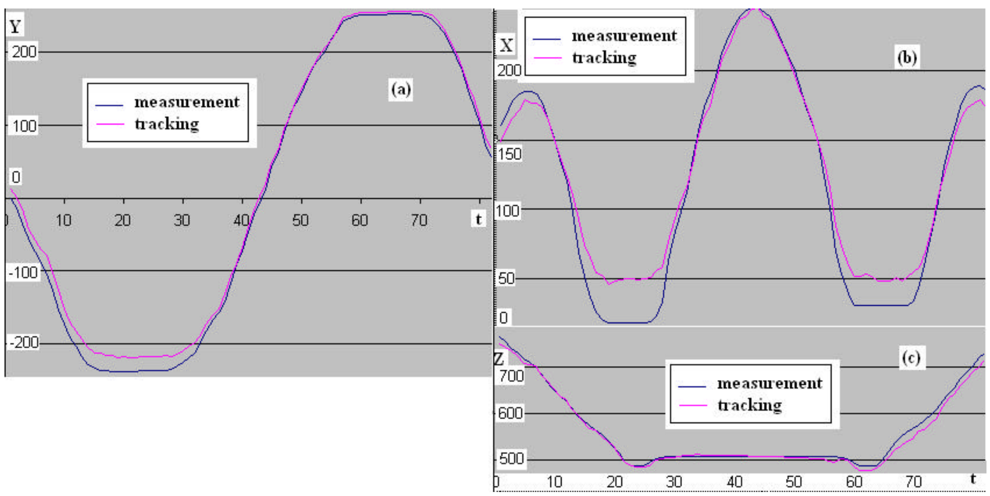


Figure 8.2**:** Tracking position with respect to the X, Y and Z-axes.

8.1.4. DISCUSSION AND CONCLUSIONS

Results of the accuracy testing yielded a very consistent error value for various locations on the work surface. The error values were location and axis specific and were no greater than twelve millimeters in some of the locations. As noted earlier, two types of error factors are of concern in this test. The first factor is that the previous accuracy testing with this system indicates a consistent and repeatable error value of up to twelve millimeters in some locations and some axes and may be a factor in the robot trajectory data. The second factor is that the tracking camera errors, if any, may be additive, subtractive, or of no effect on the errors that the robot trajectory may include

The x and y-axis coordinates exhibited the greatest error between the robot coordinates and the tracking camera coordinates from the center to the left side of the work surface as viewed from the perspective of the midpoint between the two stationary tracking cameras and facing the robot. The z-axis coordinates exhibited the greatest error between the robot coordinates and the tracking camera coordinates as the robot arm was at the home position or was traveling to or from the home position.

Both the x and y-axis coordinates error values tended to increase with distance from the REFES system camera. This error trend was the most evident at the left side of the work surface but was noticeable to a lesser degree on the right side of the work surface as viewed from the perspective of the midpoint between the two stationary tracking cameras and facing the robot.

Some of the error values have been hypothesized to be a result of the progressive distortion effects of the camera lens proportional to increasing angular displacement from the centerline of the camera lens. The results tend to support this hypothesis although further investigation is warranted.

Finally, the data from this test series indicates a consistent and repeatable error value curve that is primarily location specific. The error value curves from all of the trajectory iterations tested were very consistent and did not prove to be direction or motion sensitive. Worst-case error values exceeded the desired twenty-five millimeter error target by a maximum of ten millimeters in one of the iterations. For use in an FES application a thirty-millimeter error value may be acceptable. Since the error values are consistent with location and form a well-defined curve the transformation matrix may be able to be modified to compensate for some of the error values that were obtained in this test series. If the transformation matrix is modified to compensate for the error value patterns as were observed in this test, a repeat of test iteration one and two would be the only test required to verify the improvement in the tracking accuracy.

8.1.5. SUMMARY

This test series was designed to determine the ability of the REFES system to act as a feedback control for FES by tracking the position of the robot arm as it moves a test object between two locations on the work surface. Position coordinates were recorded simultaneously from the both the robot coordinates and from the monitoring cameras coordinates as the robot arm moved the test object. The difference in value between corresponding data points in the two sets of coordinates yielded an error value for each axis and set of locations. The average error value for each axis was well within the maximum twenty-five millimeter error value that was the desired result of this test series. The error values for all of the test iterations except for iteration ten exhibited peak error values over twenty-five millimeters for at least one axis. Graphing of the coordinates clearly illustrated that the error values were location specific with the greatest errors correlating to locations distant from the VZX camera. The x and y-axis path coordinates as graphed clearly indicate that the error values are weighed more heavily from the center of the robot work surface to the left of the robot work surface as viewed from between the stationary motion detection tracking cameras looking toward the robot. Prior accuracy tests with this system revealed an error value of up to approximately twelve millimeters for some locations on the work surface. The robot coordinates correspond to and are derived from data related to the accuracy tests and the robot coordinates are the base line for the error values for this test. In general the overall tracking of the robot arm position during the move of a test object was generally successful with no erratic coordinate data being observed in the graphed data. The maximum error values between the robot arm trajectory coordinates and the tracking camera coordinates slightly exceeded the thirty millimeters in some positions in the work envelope. The accuracy of the tracking cameras as a feedback control in a FES system may be able to tolerate a thirty-millimeter error value for some locations within the work envelope.

In summary, the basic feasibility for RobotEyes™ to provide position signals for use in neuroprosthesis control has been demonstrated. There are a number of significant advantages to such a scheme if it can be practically realized.

8.2. DEMONSTRATE HAND TRACKING

In this section, a hand tracking demonstration was performed and the accuracy of the tracking is discussed. The purpose of this demonstration is to show the capability of REFES system to extract the hand segment from image sequences captured by the 2D tracking cameras, to determine the 3D position of the moving hand as a function of time, to estimate the motion parameters and to predict the one step ahead moving position. During the demonstration, a hand model was attached to the end-effector of the Mitsubishi RV1A robot and a simple motion of the hand model was controlled by the robot. The robot was programmed to move the hand model as if it were going to pick up an object on the working table.

8.2.1. SYSTEM CONFIGURATION

The system was set up based on the configuration requirements from the REFES system with the similar set up illustrated in Figure 8.1 with out the marker. Rather than use the marker with the robot gripper, a model of a human hand is used to simulate a real human hand and the Mitsubishi RV1A robot arm was used to simulate motion of a human arm as shown in Figure 8.4. The reason a real human hand was not used during the demonstration is that the motion of a human hand practically is neither controllable and repeatable nor measurable. Obviously, while the REFES system is capable to track the motion of a real human hand, the tracking would not able to be tested or evaluated.

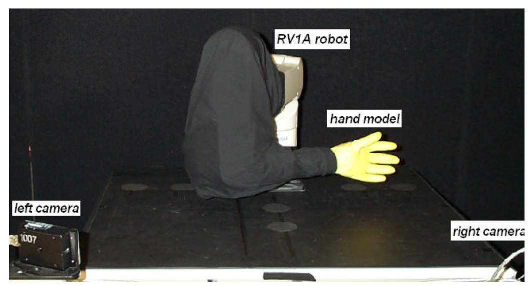


Figure 8.4: Hand motion tracking.

The installation of the hand model was designed to face both 2D tracking cameras, the "left camera" and the "right camera" in Figure 8.4. The hand model motion trajectory was designed as a simulation of a human hand moving to pick up an object from the table and guaranteed there is no blocking object between the 2D tracking cameras and the moving hand. Thus there is no motion hidden from the view. For the purpose of speeding up the color image segmentation during the experiment, the robot body and the working background were covered with dark clothes. The tracking cameras captured a sequence of images while the robot moved the hand model. Two set of images with their corresponding frame number from the 'left camera" and the "right camera" are shown in Figure 8.5. In this figure, numbers of the frames from the sequence are picked in order to see the obvious changes of the view. It shows that the image from the difference sequence provides two different views of the hand model from two tracking cameras.

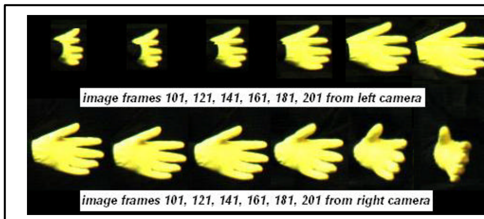


Figure 8.5: Images taken from left and right cameras with frames numbers.

Hand motion positions were tracked using the methods discussed in Section 4.2.4. Both estimated motion trajectory and desired motion trajectory are plotted in Figure 8.6 with respect to the X, Y and Z Axes. The upprer part of the figure shows that while the tracking positions follow the real motion of the hand model, a periodical error appears that corresponds to the motion of the hand with respect to the X-axis, the Y-axis and the Z-axis. This is because of the calculation errors of the hand position. When the hand moves, its orientation to the tracking cameras are changing. The change causes the change in calculation of the hand segments. When a hand segment is large enough, the calculation of the hand center from the segment is much more accurate. On the other hand, as the hand segment becomes smaller, the error to extract the center can be larger. Errors in 3D distance between tracking trajectory and real motion trajectory appear with periodical characteristics as shown in lower part of Figure 8.6.

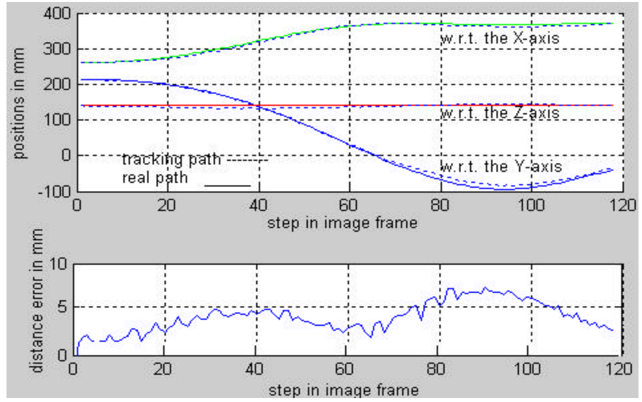


Figure 8.6: Hand position tracking trajectory compared with real motion trajectory.

Similar to its periodical characteristics with respect to different axes, the position tracking errors in 3D distance have its periodical characteristics as shown in Figure 8.7. The reason is that the estimation of the moving hand orientation is calculated based on the estimation of moving hand positions. This in fact provides an opportunity to design a filter to correct the position tracking based on change of hand orientations. While the absolute hand orientation tracking errors are still smaller than 1 degree, the error distribution is irregular as shown in the figure.

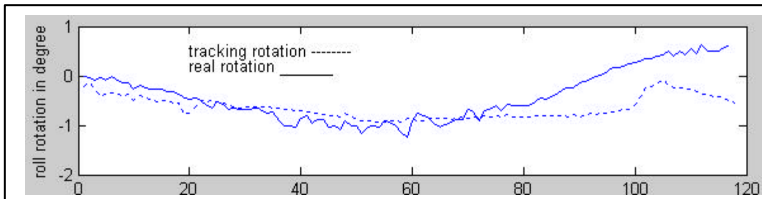


Figure 8.7: Tracking hand motion orientations compared with real hand motion orientation.

8.2.2. HAND MOTION RECOVERY AND PREDICTION

In this section, we demonstrate the ability of the recursive least squares algorithm to perform hand motion parameter estimation. A sequence of the observed 3D points from the hand motion, shown in Figure 8.6, was used as the observed values for equation (4.7). Values as large as 10^{12} are used for the initial values of the covariance matrix P_0. The initial estimate parameter vector $\hat{q}_0$ is set to $\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}'$. The initial noise matrix of equation (4.4) is initialized randomly by $c_{(1,1)} = 0.5$, $c_{(2,2)} = -0.6$ and $c_{(3,3)} = 0.5$. A set of 40 unit random white noisy points together with the parameter vector of equation (4.4) and the state matrix of equation (4.3) were used. The RLS algorithms presented by Equations (4.9), (4.10) and (4.10) are allowed to continue through 40 iterations. The results are shown in Figure 8.8.

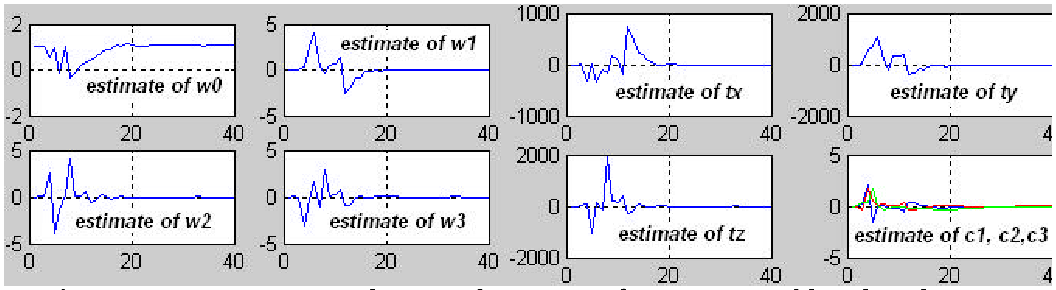


Figure 8.8: Rotation, translation and noise transformation variables plotted against iteration count of image frame number for the RLS algorithm.

The estimates of the parameter vector reach a stable value after about 20 iterations. A set of the computed estimates is obtained and the parameter estimations are plotted in Figure 8.9. The final computed estimation parameters are:

- Estimated rotation variables: 0.0053 -0.0063 -0.0008
- Estimated translation variables: 4.3559 0.0528 -0.9922
- Estimated noise variables: -0.0069 -0.0878 -0.0369

A sequence of prediction of the hand motion was calculated by Equation (4.1) based on the estimate parameters. The comparison of the motion prediction with the tracking hand positions was plotted and as shown in the first part of Figure 8.9. Detailed plots of comparisons between the prediction and tracking positions with respect to different axes are also shown in lower part of figure. While the observation (tracking) of the motion appears with a large portion of noise, the prediction was generated from equation (4.1) without a noise term and shown a smooth motion that can be closer to the real motion. This large portion of the moving noise in the observation is in fact caused by the image segmentation and calculation of center of the hand segments.

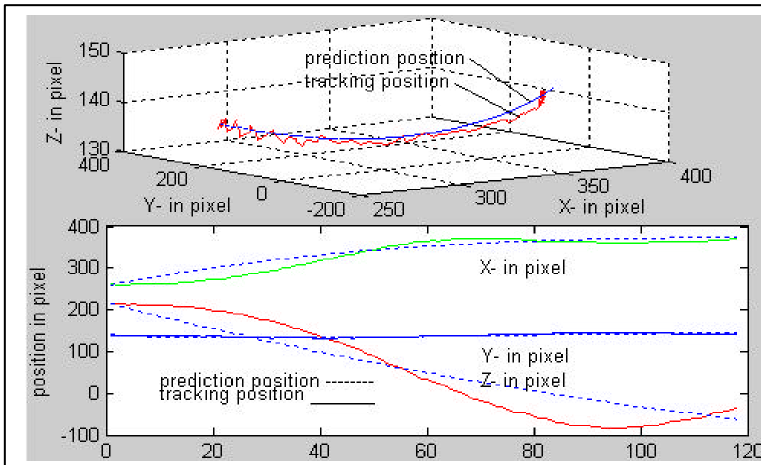


Figure 8.9: Hand position prediction and tracking comparison in 3D space and comparison with respect to all axes.

The test and demonstration results indicate that the tracking positions follow the measured position closely while a consistent and repeatable error value curve that is primarily location specific. Worst-case error values exceeded the desired 25-millimeter error target by a maximum of 10-millimeter in one of the iterations. For use in a FES application a 30-millimeter error value may be acceptable. The error value curves from all of the trajectory iterations tested were very consistent and did not prove to be direction or motion sensitive. Since the error values are consistent with location and form a well-defined curve the transformation matrix may be able to be modified to compensate for some of the error values that were obtained in this test series. If the transformation matrix is modified to compensate for the error value patterns as were observed in this test, a repeat of test iteration one and two would be the only test required to verify the improvement in the tracking accuracy.

This will assess the ability to interpret the environment and recognize objects using the same basic setup as the test one. The manipulator will be programmed to point to a location provided by the REFES system and RE will be set up to track a recognizable object. The evaluation team will then move the object within the workspace and assess the ability to recognized the object and perform action with the object (picking up a cup, moving it, etc.). The tracked arm motions will also provide feedback to the arm control, demonstrating the ability of the arm to accurately pick up and place desktop objects even with erratic or perturbed arm motion.

8.3. SUMMARY

In summary, the results from tests and demonstration in robot and hand motion tracking either with marker or without marker show that the REFES system has a capability to track the motion of a know object, either a robot gripper or a human hand. While the average of tracking error is no larger than 8 mm over a consistent and repeatable error value curve, few errors can be larger than 25 mm of tracking limitation. The image captured speed for matching is larger than 20 frames per second. The demonstration also shows the motion prediction using a fast prediction speed of 1 second.

The tracking speed and prediction speed satisfy the matching requirement of a sufficient rate for neuroprosthesis applications. While the orientation tracking accuracy need to be improved, the position tracking accuracy is not larger the maximum allowed error of 25 mm. We conclude that the REFES system is capable of providing position feedback control for neuroprosthesis applications.

9. CONCLUSIONS AND FUTURE WORK

9.1. CONCLUSIONS

During this phase, the REFES System was enhanced and evolved into an intelligent vision based system. The system has capabilities in image capture and processing within a related robot's working environment. The working environment can be a fixed working table, or a platform with in site condition. The system is capable of:

- Reconstructing certain type of 3D objects
- Capturing the 3D scene of the working environment
- Gathering and processing 3D information and knowledge about the objects and the working environment
- Understanding this information and knowledge and using it to monitor the changes in the working environment
- Operating on a set of objects by controlling a robot arm with collision free moment
- Tracking motion of a known object, such as a human hand or a robot end-effecter.

The successful development of the REFES system during this phase, is in fact, a result of applications of a collection of advanced technologies that include real time image capture and processing, 3D surface reconstruction, 3D modeling and target recognition, camera calibration, robot control, intelligent trajectory planning, obstacle identification and avoidance, dynamic system identification, motion recovery and prediction, and position feedback control.

The capabilities of the REFES System provide an effective user interface and optical spatial feedback controller for a neuroprosthesis for individuals with high tetraplegia resulting from high cervical spinal cord injury. The system also provides other related applications, such as the command interfaces for rehabilitation robots that are commonly used by individuals with high tetraplegia, muscular dystrophy, amyotrophic lateral sclerosis (i.e., "Lou Gehrig's disease"), and other neurological and musculoskeletal disease.

The demonstrated capabilities of the REFES System are based on algorithms and functions that have been developed and integrated into the system successfully during this phase, include:

1 2D and 3D image captures and processing: Algorithms and functions in real time 2D image capture and record, image segmentation, camera calibrations, image distortion, and image color component processing.
2 3D object and scene reconstruction: Algorithms and functions in edge detection; noise filtering and 3D point generation.
3 3D knowledge gathering and processing: Algorithms and functions in 3D object property definition, 3D object identification, 3D surface matching, 3D object separation, etc.
4 3D working environment understanding and intelligent management: transformation from 4. 5. 2D camera coordinate to robot coordinate, transformation from 3D camera coordinate to robot coordinate, robot workspace design, 3D matrix design, robot forward and backward kinematics development, robot trajectory planning, etc.
5 Real time 3D working environment monitoring: algorithms and functions in motion tracking and identification.

6 Real time motion recovery, prediction and control: algorithms and functions in robot controller design, robot interface design, motion model creation, motion parameter estimation, motion prediction and control, etc.

Based on the REFES system performance during a series of system tests performed at SIS and the FES Center in CWRU, Ardiem Medical, Inc. (the program's testing agent) concluded that "the REFES System is sufficiently accurate, functional, and appropriate to be interfaced with a with a prosthetic arm or limb, functional electrical stimulation device, other prosthetic or assistive device. The accuracy and performance level of the system as tested were deemed currently satisfactory for the intended purpose.

The system test performed by Ardiem Medical, Inc. also shows that the REFES system provides following accuracy properties:

1 3D observation accuracy:
   Average location identification error: 1.6-millimeter;
   Average worst-case axis error 8.68-millimeter.
2 3D tracking accuracy:
   Average tracking error: < 12-millimeter;
3 Real-time image capture rate:
   20 Hz
4 Targeting accuracy:
   Directly related to the accuracy testing results and was within the error values obtained in the accuracy tests.
5 Obstacle avoidance:
   Are overall successful within the limits of the robot workspace.
6 Hardware Interface:
   Compatible interfaces with different commercial, fully articulated robotic arms.

Based on the performance of the REFES system with the Staubli RX60B robot system in the FES Center at CWRU, the FES Center (a contractor for this project) concluded "that the REFES system is likely to play a significant role in the continued development of a neuroprosthesis for high tetraplegia. The REFES interface has several major positive attributes that are not seen in alternative command interfaces. We fully expect to implement and test the REFES interface with human subjects when real neuroprostheses are implemented in human subject in spring 2005."

"FES Center goes onto say: "These same attributes that make the REFES interface very attractive for neuroprosthesis applications to high tetraplegia also suggest several other related applications. In particular, the command interfaces for rehabilitation robots that are commonly used by individuals with high tetraplegia, muscular dystrophy, amyotrophic lateral sclerosis (i.e., Lou Gehrig's disease), and other neurological and musculoskeletal disease are currently surprisingly ineffective. Given the existing ability of REFES to control different types of robots, interfacing this system to a rehabilitation robot should be relatively straightforward."

"The additional hardware beyond the robot itself that is added to the wheelchair should be acceptable with some minor modifications to the REFES imaging system. We believe that the

REFES command interface will be far superior to existing systems and could significantly increase the market for rehabilitation robots."

9.2. FUTURE WORK

Improvements can be done to increase system accuracy to guarantee the safety implementation of REFES based feedback system. Some potential improvements are discussed here.

9.2.1 Improve camera distortion and coordinate system transformation

During the tests, it was found that the accuracy of the REFES system observed 3D location information was different from location to location. The locations close to focus of the VZX camera appear with a high accuracy. This could be caused by the image distortion. Developing more effective VZX distortion algorithms would assist in alleviating this problem. Other reason can be the transformation between the VZX camera coordinate and the robot coordinate systems. As discussion in section 2.3.1, 3D perfect models and partial 3D surface point sets generated by VZX have been used for camera calibration to find the transformations. While the generation of 3D point of a model using measurement contributes a high accuracy, 3D point generation costs a considerably large error. Errors from matching, point generation cause errors in the coordinate transformations. A better method to find the transformation can be developed without using the generated 3D point sets.

9.2.1. INCREASE HAND LOCATION ACCURACY WITH DIFFERENT HAND ORIENTATIONS

Hand orientation and operation tracking was not included during this phase. However, it would play an important role in a future development. Incorporation into a neuroprosthesis system requires a robust measurement of endpoint location from the REFES system regardless of hand orientation or whether the hand is open or closed. Current REFES system algorithms track a human hand position based on a marker and/or the human hand segment. The estimates mean endpoint position based upon skeletal marks or the hand segment, but for the algorithm based on the hand segment, the shape of the hand will change significantly when the hand is opened or closed, as well as when the orientation of the hand changes. The shape of the hand will obviously change as it opens around an object and then closes to grasp it. Furthermore, the orientation of the hand needed to acquire different objects will vary depending on the shape and orientation of the object. This functionality could be accomplished by following solutions:

1. Algorithms can be developed to improve the current hand position-tracking methods. In future development, hand orientation would be identified (see Section 9.2.3). The hand orientation with respect to robot coordinate system can be transformed to the tracking camera coordinate system. A 2D projection can find a more accurate position within the hand segment rather than the center point of the hand segment or the marker segment the current algorithm is using.
2. New algorithm can be developed to track an open hand and a closed hand differently. This is practicable because the hand operation is under controlled of FES and therefore is known. Knowledge of the open hand and close hand can be obtained off line and used in tracking.
3. New algorithms using hand features such as fingers can be developed to increase the hand position tracking accuracy.
4. Use additional features. Additional features can be used to track the hand position, for instance, different skeletal landmarks, differences color, different locations on the arm that

are less sensitive to hand orientation. Alternatively, the user could wear several objects (e.g., rings) that present a high contrast to the imaging system and have distinctive shapes that allow robust identification of their orientation. This approach is somewhat less desirable from a neuroprosthesis perspective because of the need to wear additional objects on the hand, but it may be a viable alternative if body-based imaging provides insufficient information.

9.2.2. INCREASE HAND ORIENTATION TRACKING ACCURACY

Forearm pronation-supination is the primary movement used for orienting the hand, a critical component of any function requiring hand grasp (e.g., all of the acquisition tasks described earlier). Changes in hand orientation is made in a human user through rotation of the forearm about its long axis (i.e., pronation and supination). In neuroprosthesis applications, the speed with which this movement is performed does not need to be rapid – a forearm rotation speed comparable to the speed of the rest of the arm movement would be adequate. Although forearm rotation is a critical aspect of any grasping function, it has been difficult to measure in the past for several reasons:

1. Because pronation-supination rotations occur along the long axis of the forearm, global Cartesian movement at the forearm skin surface for a given internal angular rotation is small because of the short distance. This is in contrast to other joints like the elbow whose long body segment lengths (humerus proximally and forearm distally) produce large Cartesian motions for a given joint rotation. Measurement by markers placed on the skin surface is therefore not particularly sensitive to the internal bone rotations and prone to inaccuracies.
2. To overcome the sensitivity problems described above, devices for measuring forearm orientation can use long cantilevers that mechanically magnify the Cartesian movement for a given forearm rotation. This is impractical for a neuroprothesis, however, since such devices would in appropriately interfere with activities of daily living. This is especially true for devices attached to the hand.
3. Measuring bone rotations in the forearm (i.e., the radius relative to the ulna) using markers attached to the skin is prone to errors because of large relative movements between the skin and the bones. Thus, any measurement device attached to the forearm is susceptible to slips that can introduce significant errors into the measurements. The REFES system has the potential to overcome this problem by directly imaging bony landmarks that are relatively prominent (e.g., the shape of the hand). However, the imaging procedures will need to be able to operate for different hand configurations, e.g., different degrees of opening and closing.

9.2.3. ROBUST METHOD TO RECOVER HAND POSITION

The hand can be obscured in several ways during normal operation of the REFES system, e.g., by fixed objects during movement along a trajectory, by the user's own arm, or by objects moving into the scene. Under current REFES system configuration, the hand position can be hidden from a view of a tracking camera and causes discontinue observation and lost tracking. Discontinuous or erroneous endpoint position information due to camera blockage could lead to the robot simulator moving in a dangerous manner. Furthermore, movement of the robot simulator and/or a human with a real neuroprosthesis could result in collision with objects in the workspace, with the potential for spills. Solutions to this problem could include:

1. Increase the number of tracking cameras. By setting up additional camera from a different viewpoints can increase multiple camera intersect views and reduce the un-observed sub-within the robot workspace. Viewpoints of the tracking cameras can be carefully designed after hand operation workspace is defined.
2. Redundant imaging (e.g., hand and arm; skeletal landmarks and color variations; multiple artificial markers) so that endpoint position can be estimated even if the hand is partially obscured.
3. Hand position estimation. Hidden areas from tracking camera under certain circumstances can be calculated based on the 3D environment. Algorithms can be developed to predict if a hand is moving into a hidden area. The hand position estimate program can be designed and track the hand motion based on related information, such as the control signal to move the hand, etc.
4. Increase safe operation zone. A safe operation zone is designed whenever an operation is underway. For instance, when the robot picks up an object and move over an obstacle object, a safe zone is defined between the highest point and the lowest point of the moving object. The position of a moving hand is known with a larger error without feedback from the tracking cameras. Increasing the safe zone can guarantee the operation without the collision.
5. A control algorithm that detects blockage and interrupts the movement until a valid signal is reacquired. This additional intelligence (e.g., simply freezing the movement until a valid signal is obtained or extrapolating the currently obscured position based on previous movement state) could be included in either the REFES system or in the neuroprosthesis.

REFERENCES

[1] Koivo, A. J., 1989, *Fundamentals for Control of Robotic Manipulators*, John Wiley & Sons, New York.

[2] Harden, T. A., 2002, *Minimum Distance Influence Coefficients for Obstacle Avoidance in Manipulator Motion Planning*, Ph.D. Dissertation, Dept. of Mechanical Engineering, Univ. of Texas at Austin.

[3] Ong, C. J., and Gilbert, E. G., 1998, "Robot Path Planning with Penetration Growth Distance," *J. of Robotic Systems*, Vol. 15(2), pp. 57-74.

[4] Xu, L., and Zheng, Y. F., 2000, "Real-Time Motion Planning for Personal Robots Using Primitive Motions," *Proc. of IEEE Int. Conf. on Robotics & Automation*, pp. 1414-1419.

[5] Lee, J., 1995, "A Dynamic Programming Approach to Near Minimum-Time Trajectory Planning for Two Robots," *IEEE Trans. on Robotics and Automation*, Vol. 11(1), pp. 160-164.

[6] Macfarlane, S., and Croft, E. A., 2003, "Jerk-Bounded Manipulator Trajectory Planning: Design for Real-Time Applications," *IEEE Trans. on Robotics and Automation*, Vol. 19(1), pp. 42-52.

[7] Saramago, S. F. P., and Steffen, V. Jr., 2001, "Trajectory Modeling of Robot Manipulators in the Presence of Obstacles," *J. of Optimization Theory and Applications,* Vol. 110(1), pp. 17-34.

[8] Wang, Y., and Chirikjian, G. S., 2000, "A New Potential Field Method for Robot Path Planning," *Proc. of IEEE Int. Conf. on Robotics & Automation*, pp. 977-982.

[9] Kavraki, L., 1995, "Computation of Configuration Space Obstacles Using the Fast Fourier Transform," *IEEE Trans. on Robotics and Automation*, Vol. 11(3), pp 408-413.

[10] Latombe, J. C., 1991, *Robot Motion Planning*, Kluwer Academic, Boston.

[11] Zavlangas, P. G., and Tzafestas, S. G., 2000, "Industrial Robot Navigation and Obstacle Avoidance Employing Fuzzy Logic," *J. of Intelligent and Robotic Systems*, Vol. 27, pp. 85-97.

[12] B.Moghaddam and A. Pentland, "Maximum likelihood detection of faces and hands", Proc. Of Int. Workshop on Automatic Face and Gesture Recognition, pp. 122-128, June 1995.

[13] U. Brockl-Fox: "Real-Time 3-D Interaction with up to 16 Degrees of Freedom from Monocular Video Image Flows ", International Workshop on Automatic Face and Gesture-Recognition, 1995, University of Zurich, pp. 172- 178.

[`14] J. Rehg and T. Kanade, "Visual Tracking of High DOF Articulated Structures: an Application to Human Hand Tracking", ECCVV'1994, pp.35-46, 1994.

[15] David G. Lowe, "Fitting Parameterized Three-Dimensional Models to Images", IEEE Trans. On PAMI, Volt 13, No 5, pp 441-450, 1991.

[16] M. Mochimaru and N. Yamazaki. "The Three-dimensional Measurement of Unconstrained Motion Using a Model-matching Method". ERGONOMICS, vol.37, No.3, pp. 493–510, 1994.

[17] B. Stenger, P. R. S. Mendonça, and R. Cipolla, "Model-based hand tracking using an unscented kalman filter," in Proc. British Machine Vision Conference, vol. I, (Manchester, UK), pp. 63–72, September 2001.

[18] L. Tsap, et al., "Human skin and Hand Motion Analysis from Range Image Sequences Using Nonlinear FEM", IEEE Proc. Nonrigid and Articulated Motion Workshop, pp. 80-88, 1997.

[19] John Y. A. Wang and Edward H. Adelson "Layered Representation for Motion Analysis", Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 361-366, New York; June 1993.

[20] T.S. Huang, "Image sequence analysis", Spring-Verlag Berlin Heidelberg, 1981.

[21] S. Chang, “Rigid object motion parameter estimation from a dynamic image sequence”, Ph.D. dissertation, George Mason University, Fairfax, VA, 1997.
[22] S. Chang, J. Fuller, A. Farsaie, L. Elkins, ‘Recursive least squares approach to calculate motion parameters for a moving camera”, SPIE International Symposium on Photonics Technologies for Robotics, Automation, and Manufacturing, October, 2003.
[23] J. Bruce, T. Balch, M. Veloso, “Fast and Cheap Color Image Segmentation for Interactive Robots”, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213.
[24] S. Chang, J. Fuller, A. Farsaie, L. Elkins, “Color Feature and Density Based Image Mosaicing Using Repeated Application of the ICP Algorithm’, IS&T/SPIE 16th Annual Symposium on Electronic Imaging Science and Technology, January, 2004.

Appendix I - REFES Operating Instructions

RobotEyes™ Functional Electrical Stimulation (REFES) Program

OPERATING INSTRUCTIONS

(Preliminary)

August 31, 2003

Spatial Integrated Systems, Inc.
7524 Standish Place, Suite 100
Rockville, MD 20855
301-610-7965
SIS-R-201-03

Introduction

The purpose of this document is to provide the RobotEyes™ Functional Electrical Stimulation (REFES) system operator with instruction to operate the system. Step-by-step explanation of the user interface controls and interfaces are described. Much of the programs operation is institutive and, with the aid of this document, system operation can be accomplished in a straightforward manner. Note that this document is applicable only to the Delivery One REFES system and any default parameters used here are also only applicable to this specific system.

Physical Layout

The heart of the REFES system is the Mitsubishi RV1A robot or the Ataubli RX60B robot. Because the characteristics of these two robots are somewhat different, the screens shown herein, as well as any physical representations of the robots, is representative of either robot. The term "robot" is used herein to represent either robot.

This robot is mounted at the rear center of a table measuring approximately three feet wide by two feet deep. The background, tabletop, and the robot arm with the exception of the grip are black or draped in black. Two stationary cameras are mounted one each at the two front corners of the table. The cameras are mounted approximately one foot or less away from the table and approximately six inches above the table and are angled roughly toward the center of the robot base. A third laterally movable camera is mounted directly above the stationary camera on the right side looking toward the workstation. Above the movable camera is a strobe illumination source. A computer monitor, keyboard, mouse, and computer are located on a table to the right of the workstation and are used to operate the RobotEyes™ system. It is this computer system and the operation of the REFES control program residing in that computer that is discussed in this Operating Instructions manual.

The robot workspace on the tabletop is a semicircular band approximately 180 degrees around the robot zero center point and centered on the zero point of the x axis. The inside radius of the band is approximately 204 millimeters and the outside radius is approximately 417 millimeters as measured from the 0,0 point of the robot.

There are limitations as to the physical characteristics of the objects that can be used with the REFES system. In this document, a Styrofoam cup is used as an illustration. Actual objects should have the following characteristics:

- Should be of a relative hard, unbreakable material to prevent the robot's gripper from deforming or damaging the object.
- Should be of a light color with minimal reflectance.
- Should not be highly textured or transparent.
- Should be at least 50 mm and no greater then 100 mm on the side that is perpendicular to the grippers. In the best case, the object should be circular with these same dimensions as the objects diameter.
- Should be no less than 40 mm in height.
- Should not be excessively heavy.

CAUTION

To avoid injury during system operation, personnel should refrain from placing any part of their body within the operating range of the robot.

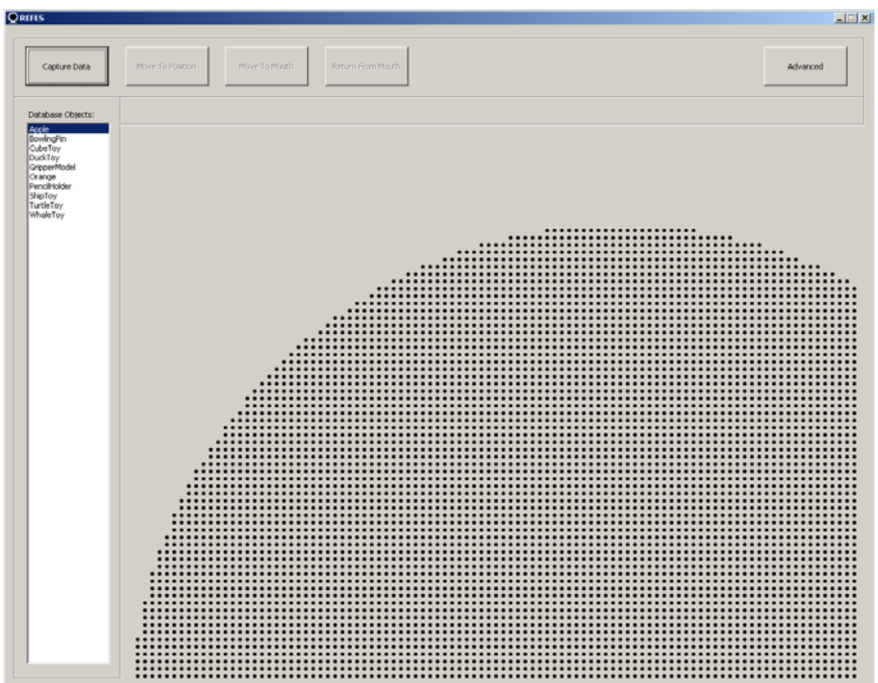
REFES System Operating Procedure

Prior to starting the REFES program, all interconnections between the control computer, the image capture systems, and the robot arm must be made and confirmed correct. With this accomplished, the computer can be started and, from the windows desktop, the icon for the REFES is selected to start the program.

The REFES, as delivered and installed, should not require the setting of system parameters or adjus™ent of the optical components. The program system settings and the setting of the lenses are completed during the installation process. Manipulating either the program settings or the lens settings following the installation calibration with out the proper instruction could cause the system to function incorrectly. A description of these system parameters is included after the operating instructions. To activate this setup process, select the **Advanced** button in the upper right corner of Screen 1.

Before proceeding, place the objects of interest, as described above, in the desired positions within the workspace. Proceed with the REFES system operation as described below. Open the program *REFES.exe* from the Windows desktop to start the program.

Step 1. Opening Screen, Operating Screen 1

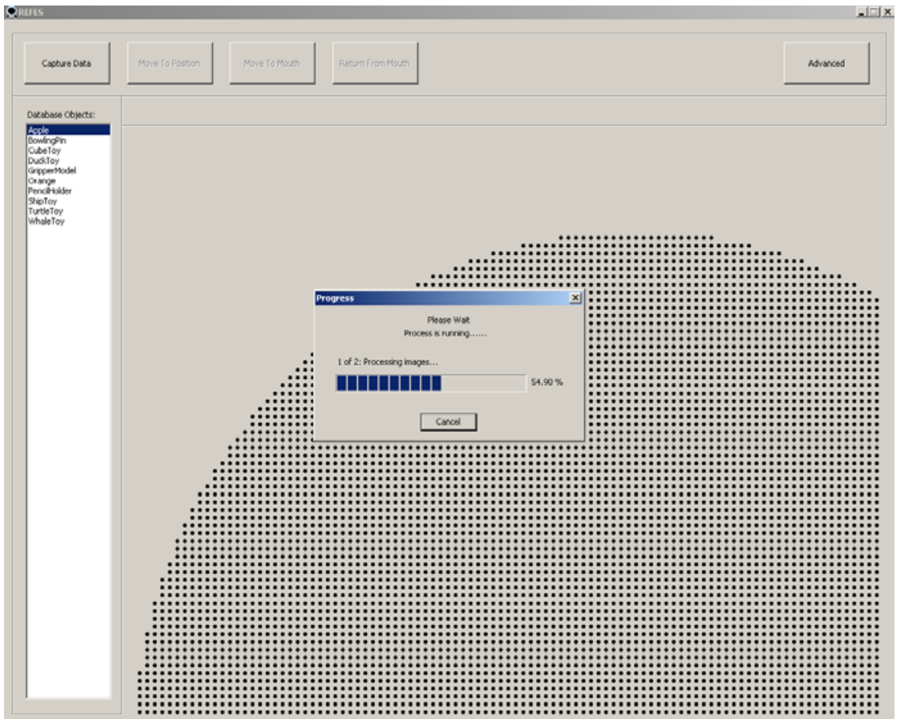


Operating Screen 1

There are three areas of note on this first opening screen. First is the pre-generated Database Objects displayed on the left of the screen. These are the names of the objects used within the workspace and are used later in the program when selecting objects to be moved. The second are the dots displayed on the main window that represent the extent of the robot workspace is 3-D. Screen 1 only show the 2-D section on the table surface, but there is an unreachable area that is due to the robot mounting post

That area is not shown in the dot pattern. The third area of interest is the **Capture Data** control button that will be used to start the date capture process, described below.

Step 2. Capturing Data, Operating Screen 2



Screen 2

The system is now ready to capture the image data to determine the object's position within the workspace. Select the **Capture Data** button in the upper left of the main window. Doing this initiates the imaging process. There are three separate processes that take place during this evolution.

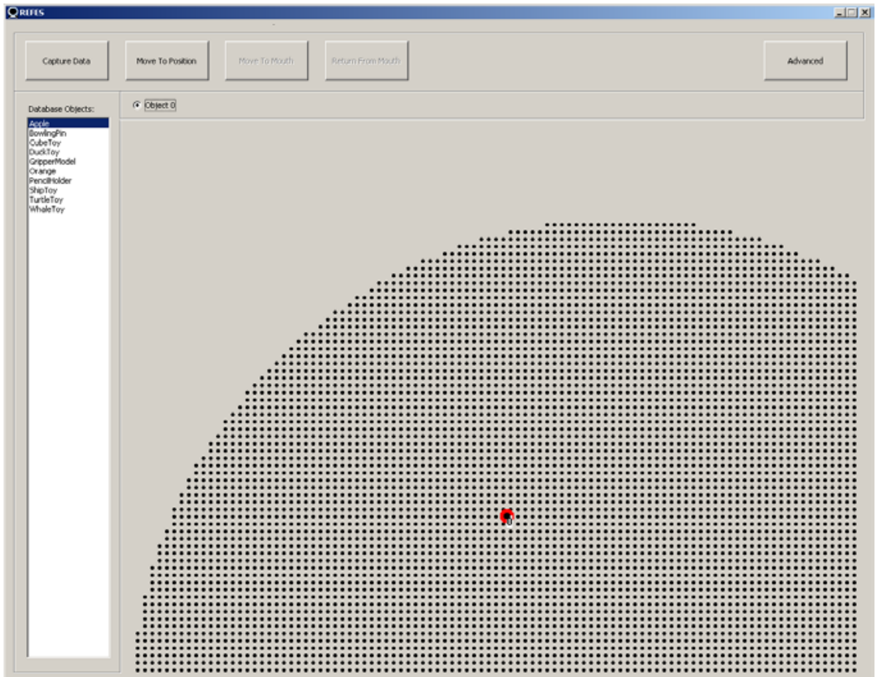
- First, the computer commands the slider motor to position the Main Camera at one end of the slider and then move it to the other end while capturing images of the object(s).

- Second, the computer processes the image data and parses it in preparation into a form compatible with the analysis process.

- Third, the processed image data is analyzed to determine the positions of the objects within the workspace.

A **Wait** window pops up and a processing bar display meter provides a graphic of the processing's progress.

Step 3. Processing Complete, Operating Screen 3

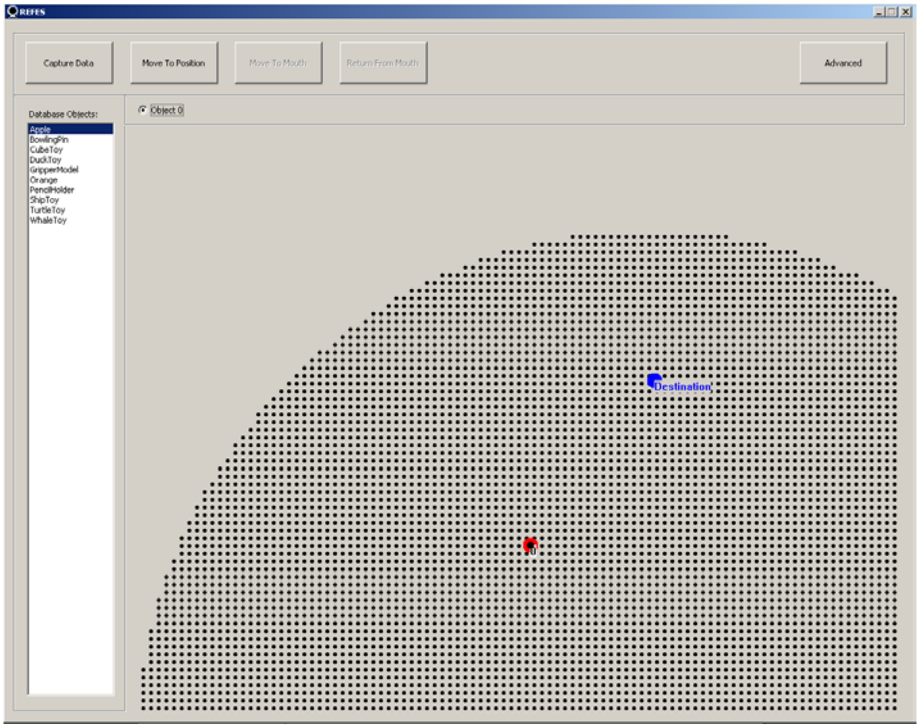


Operating Screen 3

When the processing is complete, within the workspace will be displayed a small red and black bulls eye circle(s) the represent the location of the object(s). Within this dot is a number that corresponds to the objects located at the top of the main window.

In addition, a **Move To Position** selection button is available at the top of the window to allow the selection of motions for the RobotArm to complete, as shown in the next step.

Step 4. Move To Position, Operating Screen 4



Operating Screen 4

In this screen, a blue circle with a **Destination** text attached is placed on workspace by the program. By selecting the desired object to move from the list at the top of the window, the operator can click on a location within the workspace and a blue dot will be placed on workspace at the desired location. The operator than selects the type of object from the list on the list of **Database Objects** on the left of the window. By selecting the type of object from the list, the program matches the object with its model so that it's size and orientation can be determined. With this information and the location of the blue dot in the workspace being known,, the program determines the Robot Arm's trajectory. With the selection of the object from the list, the robot arm automatically starts the sequence of moving to the object on the red dot, pickup the object, moving it to the blue dot marked location, and setting it down

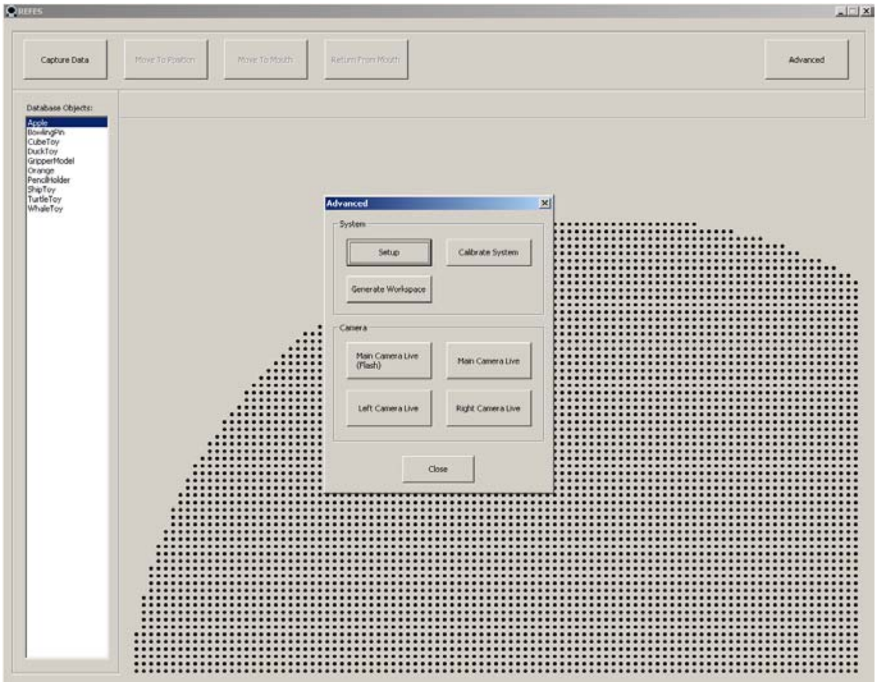
If there is more than one object, they are listed on the top of the window and are selectable from there.

Note that the object used in this instruction set is a Styrofoam cup. This was used as a representative object and is not, of course, in the **Database Objects** list. Note also that the list of object in the **Database Objects** list shown in these instructions may differ from the ones actually used

End of Operating Instructions

REFES System Setup Procedure

Step 1. System Preparation – Advanced Setup Parameters, Setup Screen 1



Setup Screen 1

From the program's startup page The **Advanced** option was selected to bring the program to this setup window**.** The **Advanced** window is now displayed and the options available. These options are:

System -

Setup –
Selecting this will bring up the System Settings window. The options presented in this window are detailed below.

Calibrate System –
This option is used to recalculate the spatial relationship between the camera and robot arm grippers.

Generate Workplace –
This option regenerates the extents of the robot's workspace based on the information that has been stored in the program's RobotConfig.txt file.

Camera -

Main Camera Live –
This selection will turn on the main camera and take sequential images of the workspace from the camera's viewpoint. A representative Main Camera Live display is shown in Screen 4.

Main

Camera Live (Flash) –
When this option is selected, the main camera will begin to take images in sync with the line generating flash. These images are again taken from the Main Cameras viewpoint. The images can be observed on the computer monitor. See Screen 5 for an example of the image from the Main Camera Live with the flash active.

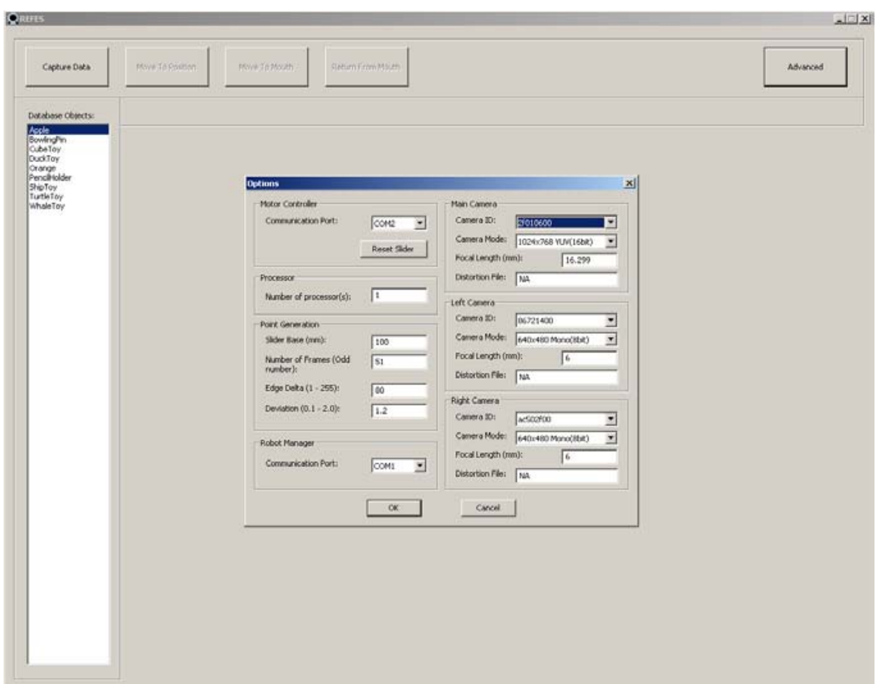
Left Camera Live-
When this option is selected, the monitor will display live images from the viewpoint of the Left Camera.

Right Camera Live-
When this option is selected, the monitor will display live images from the viewpoint of the Right Camera.

Close –
This selection closes the **Advanced** window.

To continue, the operator has two options as to how to proceed. If the system has been run prior to this current evolution and no system parameter changes are required, the operator can go right to Step 4 and the confirmation that the Main Camera is functioning. All adjustments and inputs to this window should have been made during system installation. If detailed adjustments are required to the system, they can be made in the procedure detailed in Step 3. In most operating situations, Step 3 can be omitted from the operating sequence.

Step 2. System Settings, Setup Screen 2 -



Setup Screen 2

The **System Settings** window appears, where settings can be entered that control the various aspects of the system's communication and functional operation.

Note again that for the Delivery One system the operator will not be expected to change any of the default values or parameters that have been preset into these parameters. Doing so could cause system instability. For completeness though, the six major items with their default values are as described:

Motor Controller –

Communications Port – Select the communications port to which the REFES slider motor controller is connected to the control computer. (Default value is COM2)

Reset Slider – This function moves the Main Camera to its home position in the center of the slider bar.

Processors –

Number of Processor(s) – Indicate the number of computer processors are being used in this configuration. (Default value is 1)

Points Generation –

Slider Base (mm) – Enter the length of travel of the camera on the unit's slider. (Default value is 100)

Number of Frames (Odd number) – Enter the number of images the camera is to take during the image capture sequence. (Default value is 51)

Edge Delta (1 – 255) – Enter the edge detection threshold. Note that the lower the value here, the more points are generated. (Default value is 10)

Deviation (0.1 – 2.0) – Enter. (Default value is 1.2)

RobotManager –

Communication Port - Select the communications port to which the REFES is connected to the RobotManager control computer. (Default value is COM1)

Main Camera –

Camera ID – Select the ID of the main VZX camera. This number is obtained by the program obtaining the camera's unique number from the camera itself. Each camera has a unique number and any replacement of cameras in the Delivery One system will require attention being paid to the camera IDs selection in this menu. The number selected for the REFES Delivery One Main Camera is 55010600.

Camera Mode – Select the desired camera resolution. The default value for the Delivery One Main Camera system is 1024 x 768 YUV (16 bit).

Focal Length (mm) – Enter the focal length of the lens used on the main camera. For the Delivery One system, this is 6.15. This number determined during factory calibration of the RobotEyes™ imaging system.

Distortion File – Enter the default location for the file that contains the distortion correction factors for the Main Camera lens. This file must be located in the same directory as is the executable program. Left blank or with a NA in the entry field indicates that no distortion file is being used.

Left Camera –

Camera ID – Enter the ID of the left VZX camera. Each camera has a unique number and any replacement of cameras in the Delivery One system will require attention being paid to the camera IDs selection in this menu. The number entered here for the REFES Delivery One system Left Camera is C8512f00.

Camera Mode – Select the desired camera resolution. Each camera has a unique number and any replacement of cameras in the Delivery One system will require attention being paid to the selection s presented here. The default value for the Delivery One Left Camera system is 640 x 840 Mono (8bit)

Focal Length (mm) – Enter the focal length of the lens used on the Left Camera camera. For the Delivery One system, this is 4. This number determined during factory calibration of the RobotEyes™ imaging system.

Distortion File – Enter the default location for the file that contains the distortion correction factors for the Left Camera lens. This file must be located in the same directory as is the executable program. Left blank or with a NA in the entry field indicates that no distortion file is being used.

Right Camera –

Camera ID – Enter the ID of the Right VZX Camera. Each camera has a unique number and any replacement of cameras in the Delivery One system will require attention being paid to the camera IDs selection in this menu. The number entered here for the REFES Delivery One system Right Camera is c9512f00

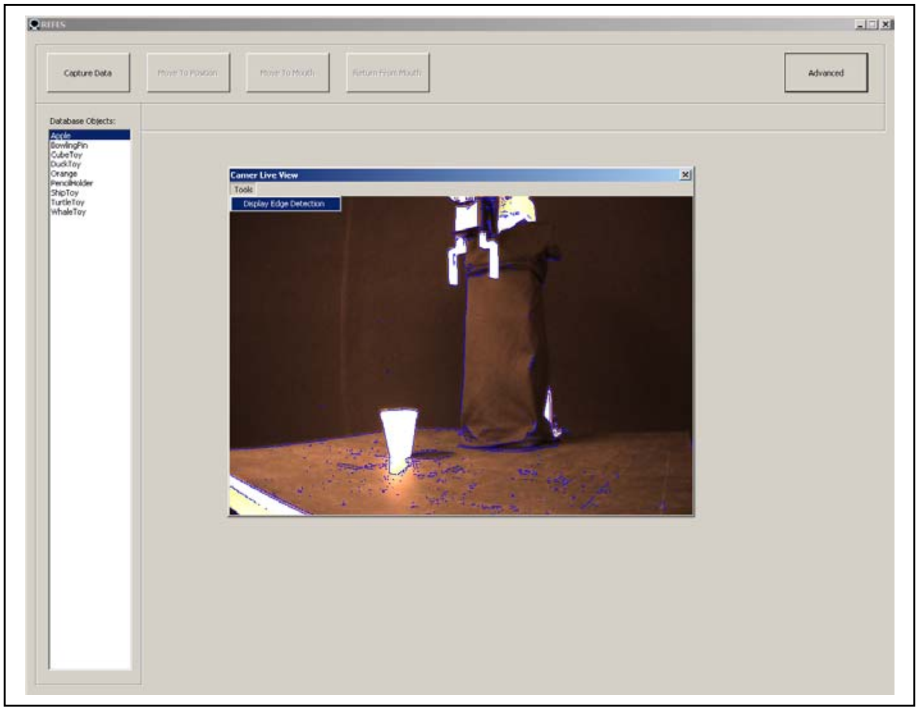
Camera Mode – Select the desired camera resolution. The default value for the Delivery One Right Camera system is 1024 x 768 YUV (16 bit)

Focal Length (mm) – Enter the focal length of the lens used on the main camera. For the Delivery One system, this is 4. This number determined during factory calibration of the VZX imaging system.

Distortion File – Enter the default location for the file that contains the distortion correction factors for the Right Camera lens. This file must be located in the same directory as is the executable program. Left blank or with a NA in the entry field indicates that no distortion file is being used.

When finished with these entries, select **OK.**

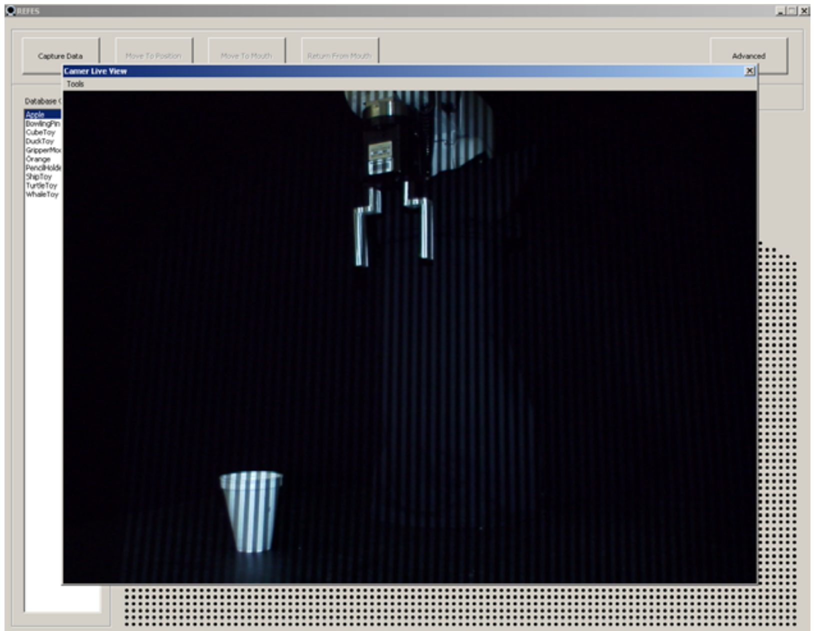
Step 3. Main Camera Live View, Setup Screen 3



Setup Screen 3

The screen here shows a representative image taken with the main camera with just the benefit of ambient light. A cup and the grippers of the Robot Arm are visible. An evenness of lighting is observed because of characteristics of the overhead lighting. This viewing option is useful only to confirm placement of the object are within the available workspace. Similarly, the images from the Left Camera and the Right Camera can be selected from the **Advanced** menu.

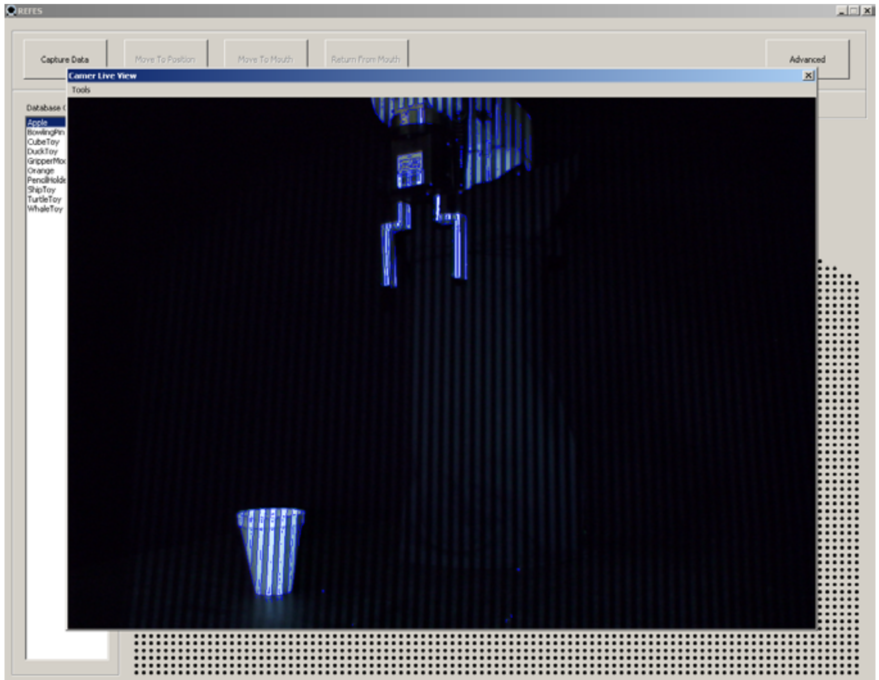
Step 4. Main Camera Live (With Flash) View, Setup Screen 4



Setup Screen 4

The screen here shows a representative image taken with the main camera along with the use of the Line Generating Flash. A cup and the grippers of the robot arm are again visible, but with the vertical lines shown on the object and the grippers. This viewing option is useful to confirm placement of the object are within the available scene, the flash / camera imaging system is working properly, and the lines are being projected on the objects.

Step 5. Edge Detection Option in Camera Live View (With Flash) Screen, Setup Screen 5



Setup Screen 5

There is an option with this screen that confirms that the imaging processing is receiving enough points to successfully process the data. Selecting the **Tools** option in the upper left corner of this Camera Live View screen activates this option. From the **Tools** drop down menu, select the **Display Edge Detection** option. The blue dots that appear on the projected vertical lines are a computer-generated representation of where the program has detected an edge. The edges of the displayed vertical lines should be densely populated with these blue dots to facilitate proper processing.

End of Setup Instructions

Appendix II - RobotEyes™ Functional Electrical Stimulation (REFES) and Robot MasterController (MC) Interface

This document is to facilitate the development of an interface between RobotEyes™ Functional Electrical Stimulation (REFES) system and an external executive device of a robot arm. REFES system provides 3D environment based on 3D spatial information of objects captured by digital cameras and operations within the 3D environments based on user inputs.

RobotManager and Interface with REFES
A logical object of RobotManager is created to represent the interface. A set of generic robot-independent control commands from REFES system, REFES Commands, is designed. These commands are based on the 3D information of robot workspace and instructions from user's interface.

RobotManager Interfaces with Robot MC
RobotManager interfaces with a robot arm by communicate with a logical object Robot MC. MC is capable of real-time motion control of the robot arm. It receives and translates of REFES commands into robot motion control commands and sends required feedback to REFES.

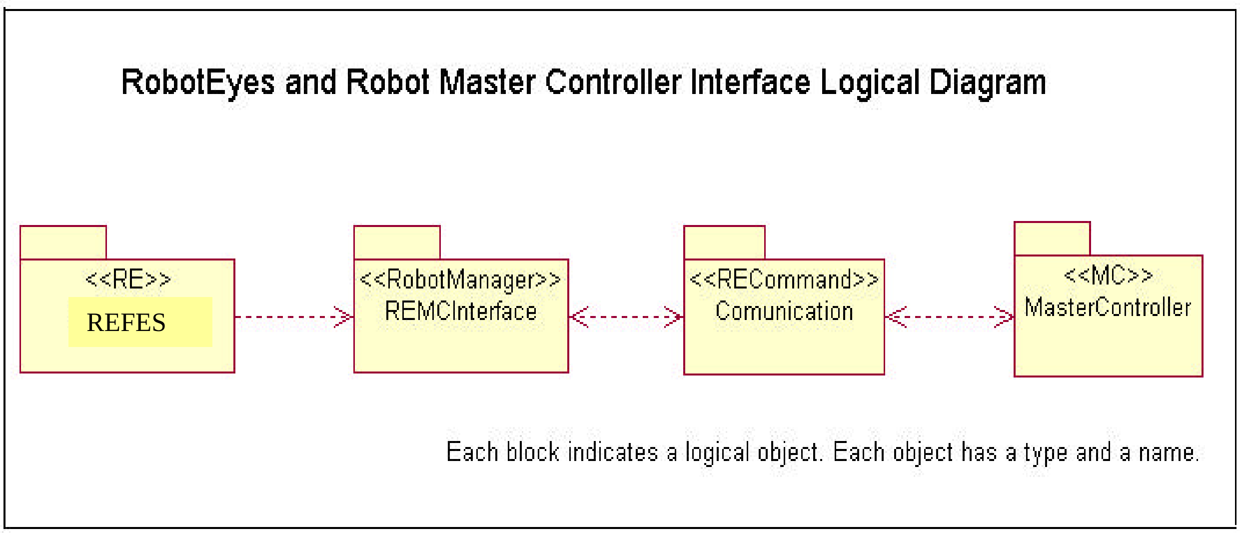


Figure 1: REFES and MC Interface Logical Diagram

A list of REFES commands and their detailed descriptions is shown in following table. Please note: one command (GET_POS) is designed specifically for robots with 6 DOF. A sample dialog between REFES and the MC is also included below. A sample dialog between REFES and the MC is also included below.

Table: List of REFES Commands

Use of this table: All REFES commands are listed in the keyword column, in uppercase. The command parameters follow the keyword, and are in lowercase. *Italicized* words are explained in details at the end of this table. Note: all keywords and parameters are separated by spaces. All replies must be trailed by a space.

No.	Keyword	Descriptions	Remarks
1	STARTUP	Initializes the *robot*.	This may include opening a communication port and turning the robot servo power on and move robot arm to home in order for the robot to get ready for future tasks.
2	HOME speed	Moves the robot to its *home position* at a given speed in *robot joint space*.	Unit: speed is in mm/s.
3	MOVE x y z a ß ? speed	Move the robot from its current position to a point with an orientation defined by (*x*, *y*, *z*, *a*, *ß*, *?*) at a given speed in *robot joint space*.	Units: (x, y, z) are in mm, (a, ß, ?) are in degrees, speed is in mm/s.
4	OPEN_GRIPPER	Fully opens the robot gripper.	MC shall make sure the gripper is fully open.
5	CLOSE_GRIPPER force	Closes the gripper and grips the object with the designated force.	MC shall make sure the gripper is properly closed. Unit: force is in Newtons.
6	PAUSE	Stops the robot's movement, but keeps the unfinished path queued.	This is used when an obstacle appears in the robot's planned path.
7	RESUME	Continues the robot's unfinished path after a PAUSE.	After a PAUSE is issued, if the obstacle moves away in a short period of time (to be determined by REFES), the robot will continue its planned path.
8	CLEAR_PATH	Clears the unfinished path, and waits for further instruction.	This command will be issued when a new path has to be generated to avoid obstacles.

Table : List of REFES Commands (cont.)

No.	Keyword	Descriptions	Remarks
9	PATH_BEGIN	Signifies the beginning of a path.	MC shall queue the commands following PATH_BEGIN as a robot path.
10	PATH_END	Signifies the ending of a path.	Once this command is received, no more actions should be queued (until next PATH_BEGIN).
11	EXECUTE_PATH	Tells MC to begin its execution of the queued path. If no queue exists then do nothing.	MC shall record the time at which it receives this command.
12	SHUTDOWN	Turns off MC. No further instructions will be given by REFES.	This may include turning off robot servo power and any desired cleanup operations.
13	SEND_POS x y z a ß ? d	Provides the visually tracked position of the arm, defined by (*x*, *y*, *z*, *a*, *ß*, *?*). The last parameter (d) is the time at which the position was visually obtained, in reference to the EXECUTE_PATH start time.	This will be used for MC to correct any tracking error from the planned path of the robot. This command will continuously be sent after EXECUTE_PATH, until a PATH_COMPLETE reply is received. Units: (x, y, z) are in mm, (a, ß, ?) are in degrees, time delay is in ms.
14	GET_POS	Acquires the *manipulator*'s joint displacements.	A *position reply* must be sent from the MC to REFES.

MC Replies:

MC shall send a *Generic* reply each time it receives a command. There are three exceptions:

1) When the robot has completed its queued path, MC shall send a *Path Complete* reply;
2) When MC receives a GET_POS command, it must send a *Position* reply;
3) SEND_POSITION does not require a reply.

The following are the types of possible replies after a command is sent to the MC.
Generic reply: "OK " (Usage example: "OK[SPACE]")
Position reply: "POS J1 J2 J3 J4 J5 J6 " - POS, followed 6 floating point values, between -360.0 and 360.0, separated by spaces.
Path Complete reply: "PATH_COMPLETE "

Appendix II: REFES and Robot MasterController Interface

Terminologies and definitions:

- **Manipulator**: the mechanical component that performs physical tasks and normally contains joints, motors, links, grippers, etc. A crane can be considered a manipulator.
- **Robot Controller**: the hardware that generates low-level commands (e.g. voltages) to the motors of the manipulator. Normally it is a self-contained box.
- **Robot**: is defined by the manipulator and the controller, which usually comes as a package from a manufacturer. A normal crane is not a robot because it does not have a controller, and only works with go/no-go status.
- **Robot Control**: the definition of this term can be fuzzy and broad in literature. Herein, it means the privilege to instruct the robot to perform certain tasks.
- **Robot Joint Space**: the space in which a robot moves with joint interpolation, i.e. a curved path. A move in robot joint space can always be accomplishable given that such a move is within robot operating range.
- **Robot Task Space**: the space in which a robot moves with linear interpolation, i.e. a straight path. A move in robot task space may not be necessarily accomplishable even though such a move is within robot operating range. This is due to the complexity of robot inverse kinematics.
- **Robot Home Position**: an initial position that is normally set by manufacturer by means of limit switches or absolute encoders.
- **(x, y, z)**: 3 translational coordinates in robot base frame that defines the location of the center of robot gripper tip.
- **(a, ß, ?)**: 3 rotational angles (roll, pitch, yaw) in robot base frame that defines the orientation of robot gripper.

NOTE: All command parameters are floating points. Speed, force and time delay shall all be positive. (x, y, z) must be within the robot operating range, and (a, ß, ?) are from -360 to 360.

An example dialog between REFES and MC through the interface:

REFES Commands	MC Replies	Remarks
STARTUP	OK	
PATH_BEGIN	OK	
HOME 200	OK	
OPEN_GRIPPER	OK	Make sure robot gripper is open.
MOVE 100 200 300 45 90 0 150	OK	Robot moves to a new location and then stops at the end.
CLOSE_GRIPPER 10	OK	Closes robot gripper with 10 N.
MOVE 400 500 600 60 90 0 150	OK	Robot moves to a new location and then stops at the end.
OPEN_GRIPPER	OK	
PATH_END	OK	MC shall not queue any more commands.

EXECUTE_PATH	OK	Robot starts to move continuously based on above path.
SEND_POSITION x y z…	----	
… (time passes)	PATH_COMPLETE	
SHUTDOWN	OK	

Appendix III - VZX Accuracy Report

A simple test to measure VZX accuracy is performed. The 3D model reconstruction accuracy outcomes are shown as follow:

1. Overall 3D model reconstruction error: 1.84%.
2. Overall vertical reconstruction error: 1.2633%.
3. Overall horizontal reconstruction error: 2.71%.
4. Overall Maximum reconstruction error: 2.78%.

Test Model:

A part model show in Figure 1 is used for the test. The measurements of the part model are shown in Figure two to compare with the 3D point model measurements.

Figure 1: Part model used for VZX accuracy test.

Camera 2D image capture:

The VZX 2D image capture is set up as follow:

ImageWidth: 1280 pixels.
ImageHeight: 1024 pixels.
Focal Length: 16.298000 mm.
Image Pixel Width: 0.0067 mm.
Image Pixel Height: 0.0067 mm.
Slider Base: 100.000008 mm.
Frame Number: 51.

VZX Point Generation:

Sub-pixel edge detection and 3D point generation is used. Part 3D point model and errors compared with measurements from part model in Figure 1 are shown in Figure 2. In each size shown in the figure, two reference numbers are displayed. The number starts with letter “M” indicates the physical measurement from the part model shown in figure 1. The number starts with letters “er” indicates the 3D model reconstruction error compared with the measurement. For instance, “M:196.2 er: 1.43%” indicates “|(196.2 – 193.39)|/196.2 * 100 = 1.43”.

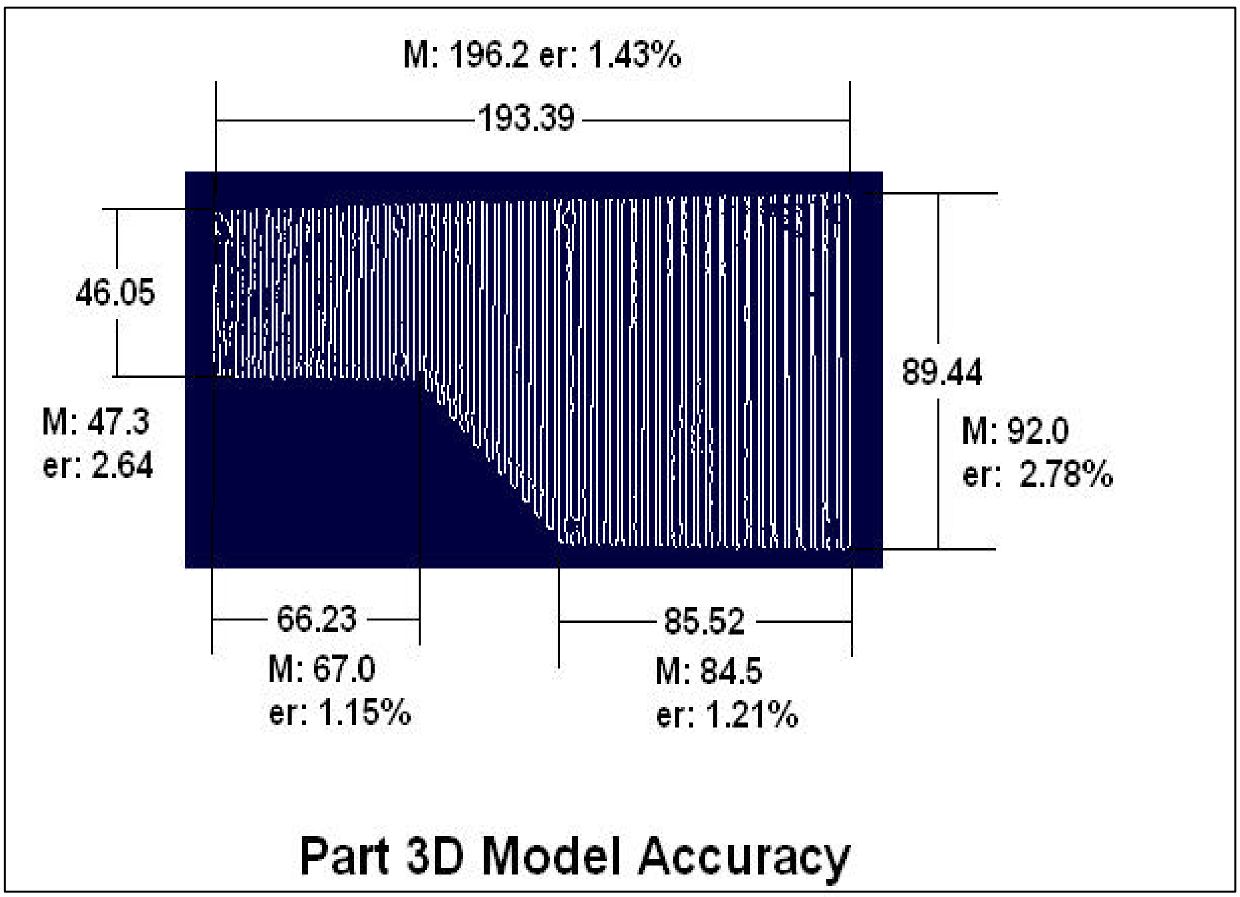


Figure 2. VZX 3D model reconstruction accuracy

Appendix IV - System Logical Diagrams

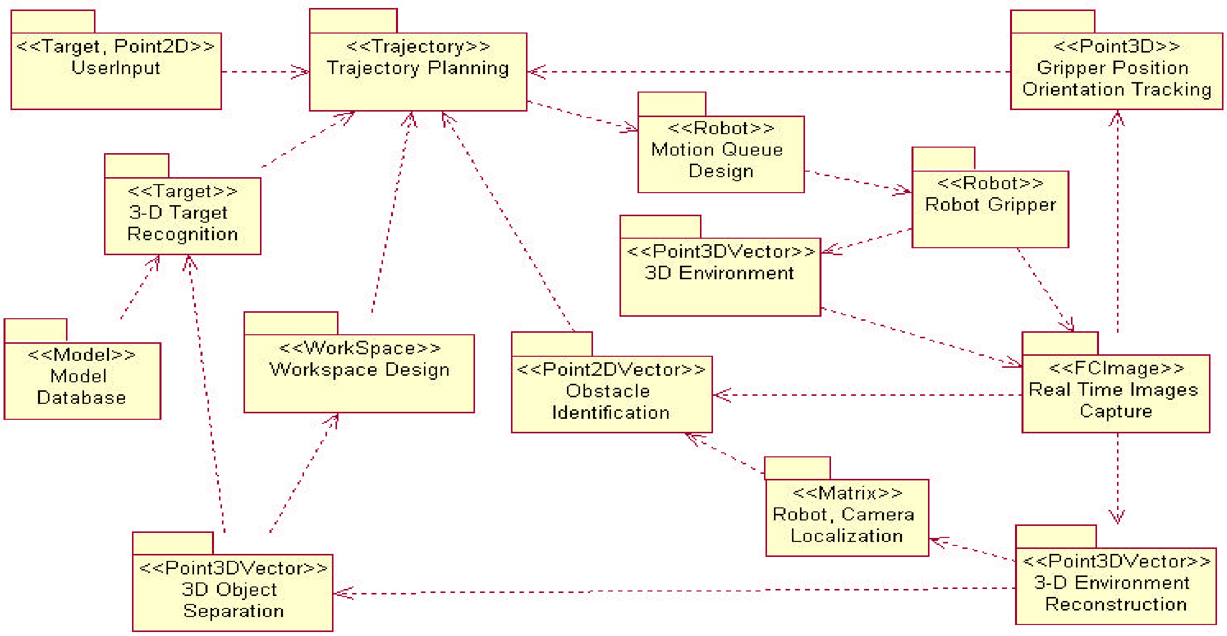


Figure A4.1; Overall system logical diagram

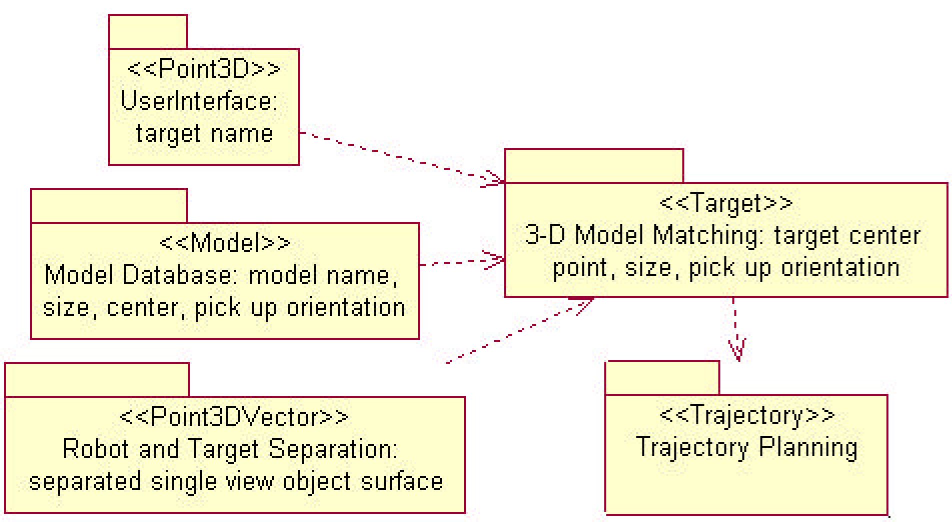


Figure A4.2: 3D model matching logical diagram

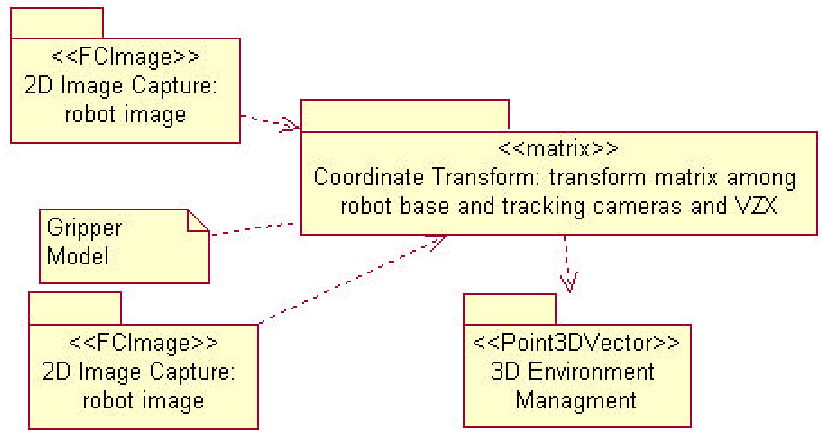


Figure A4.3: Camera and robot coordinate system transformation logical diagram

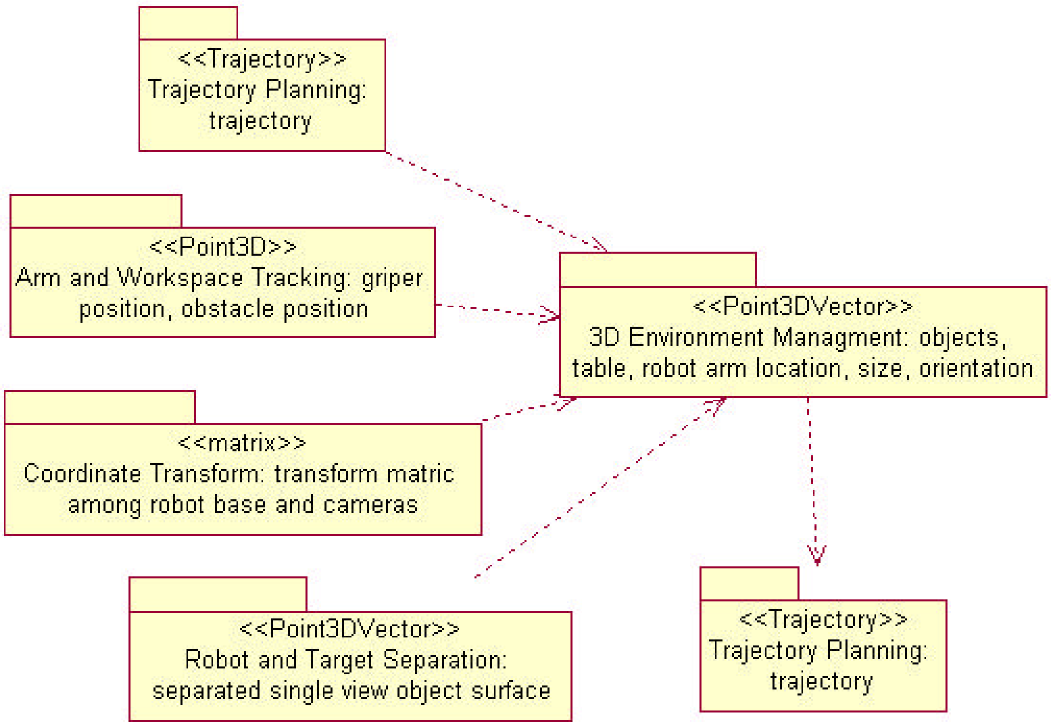


Figure A4.4 3D environment management logical diagram

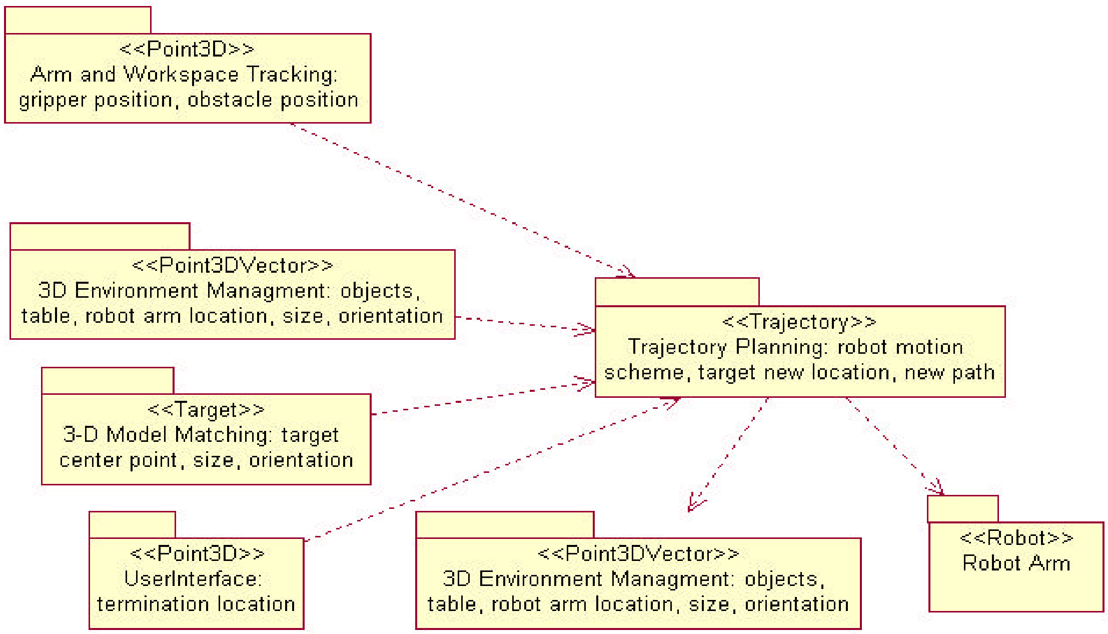


Figure A14.5: Trajectory planning logical diagram

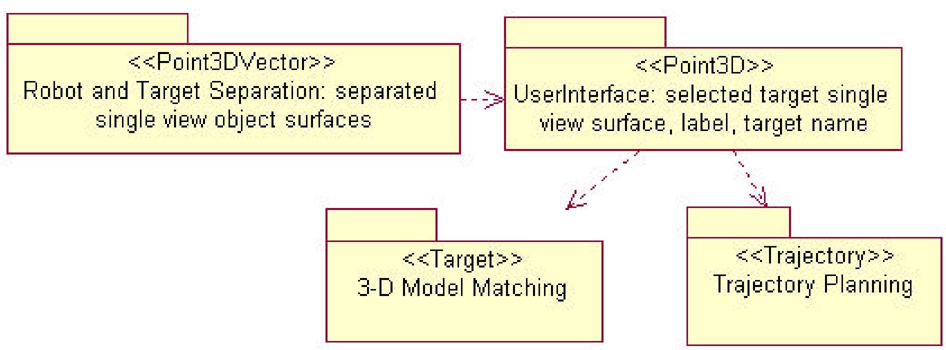


Figure A4.6: User interface logical diagram.

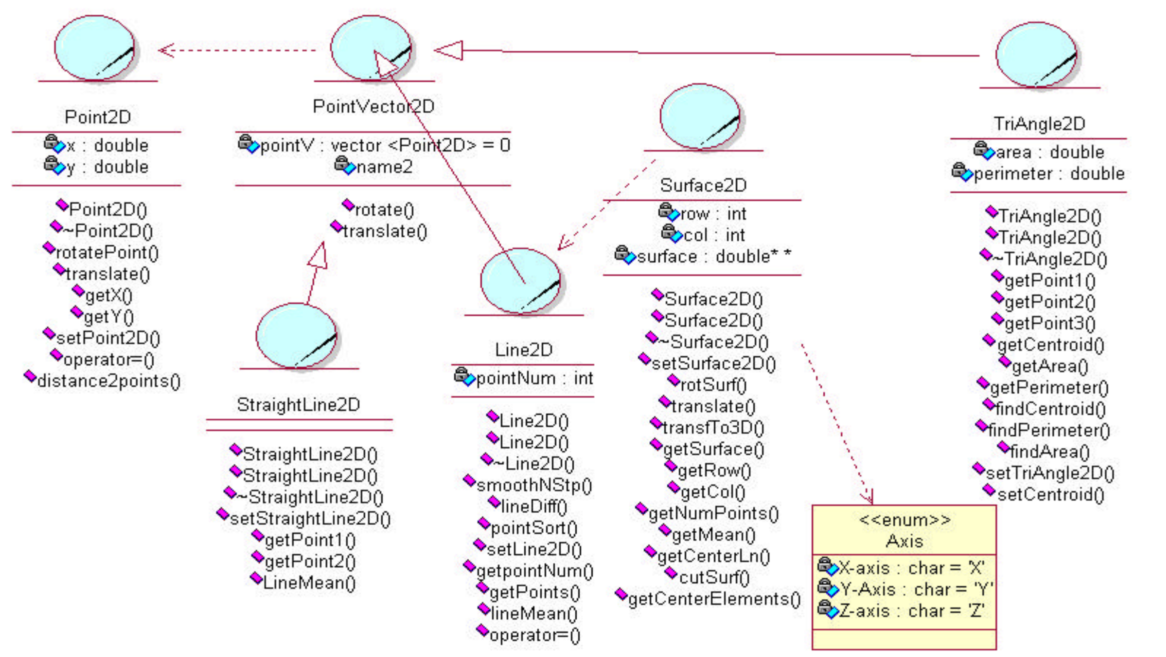


Figure A4.7: Basic 2D class design

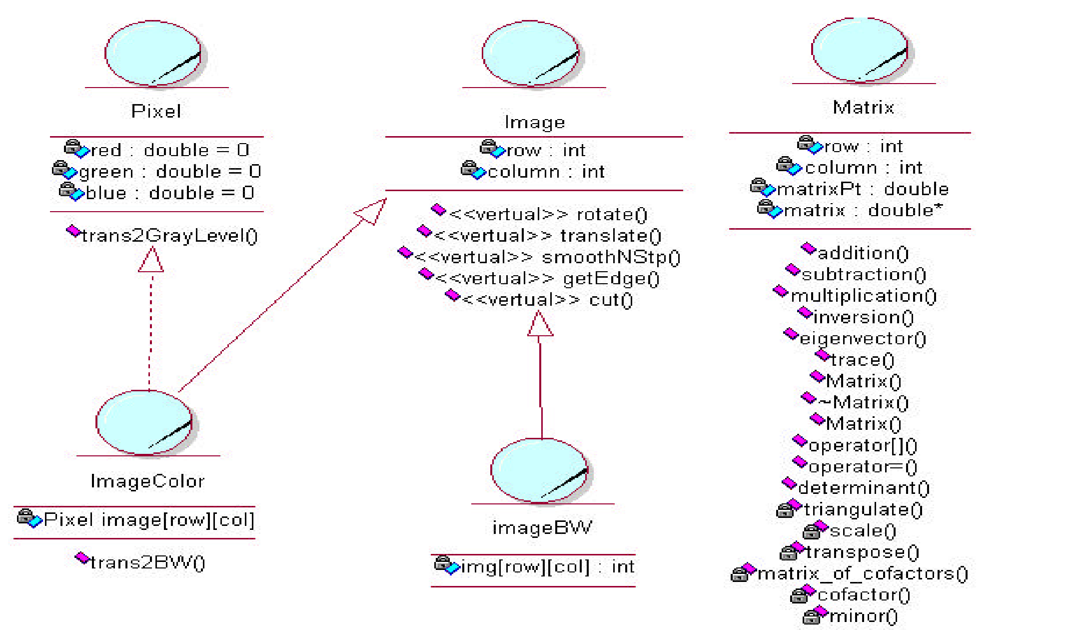


Figure A4.8: image and matrix class design

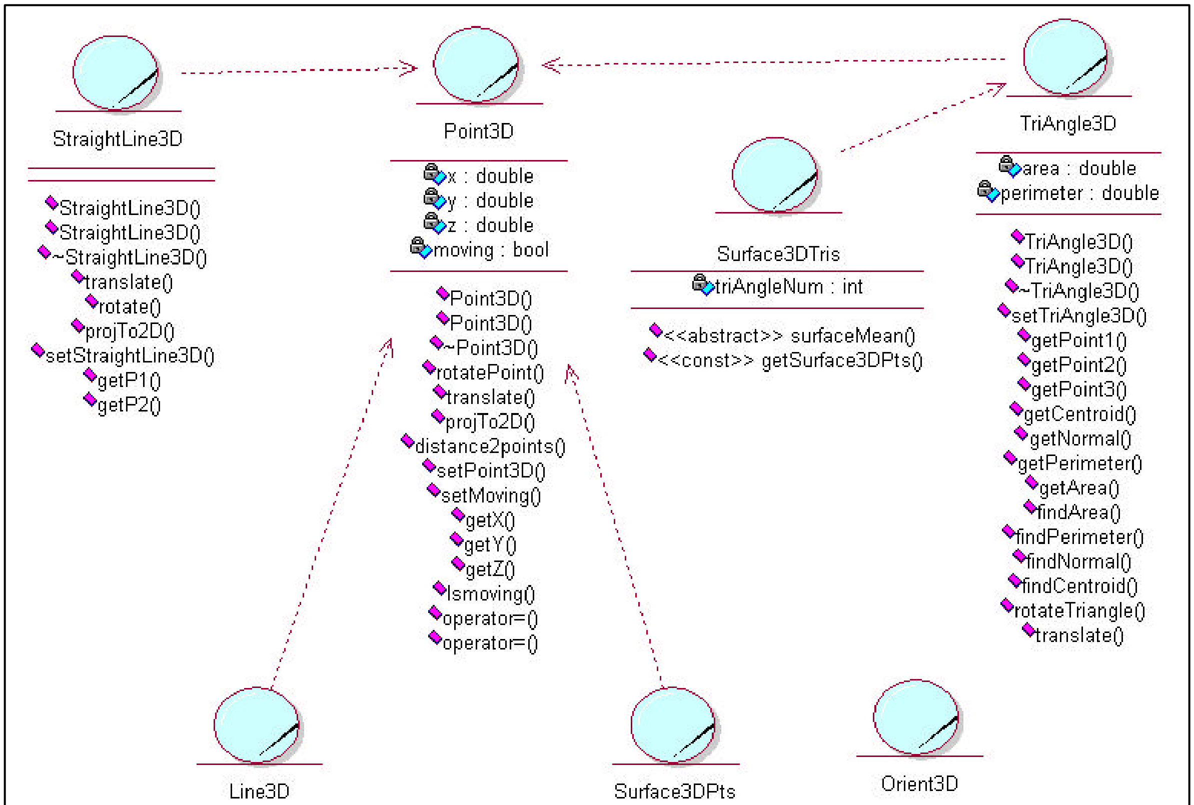


Figure A4.9: Basic 3D classes design

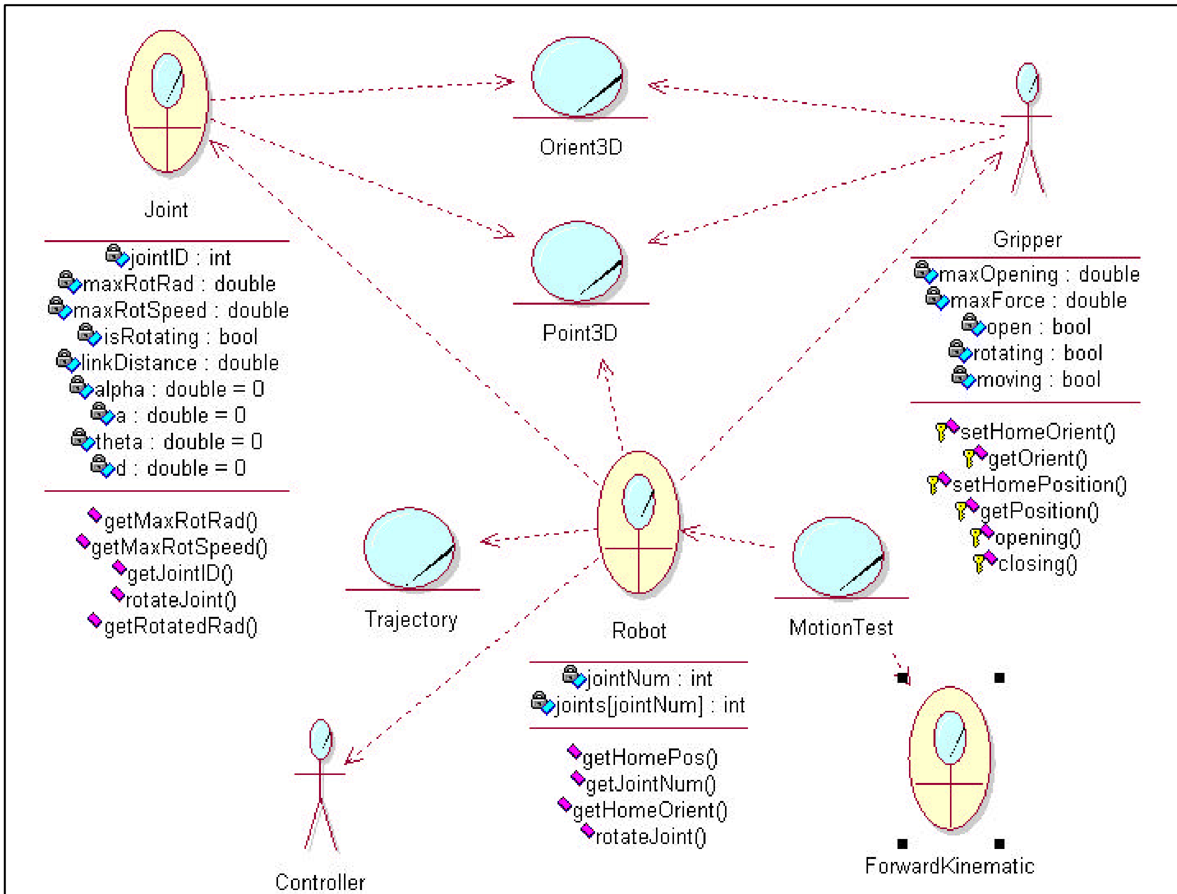


Figure A4.10: Robot classes design

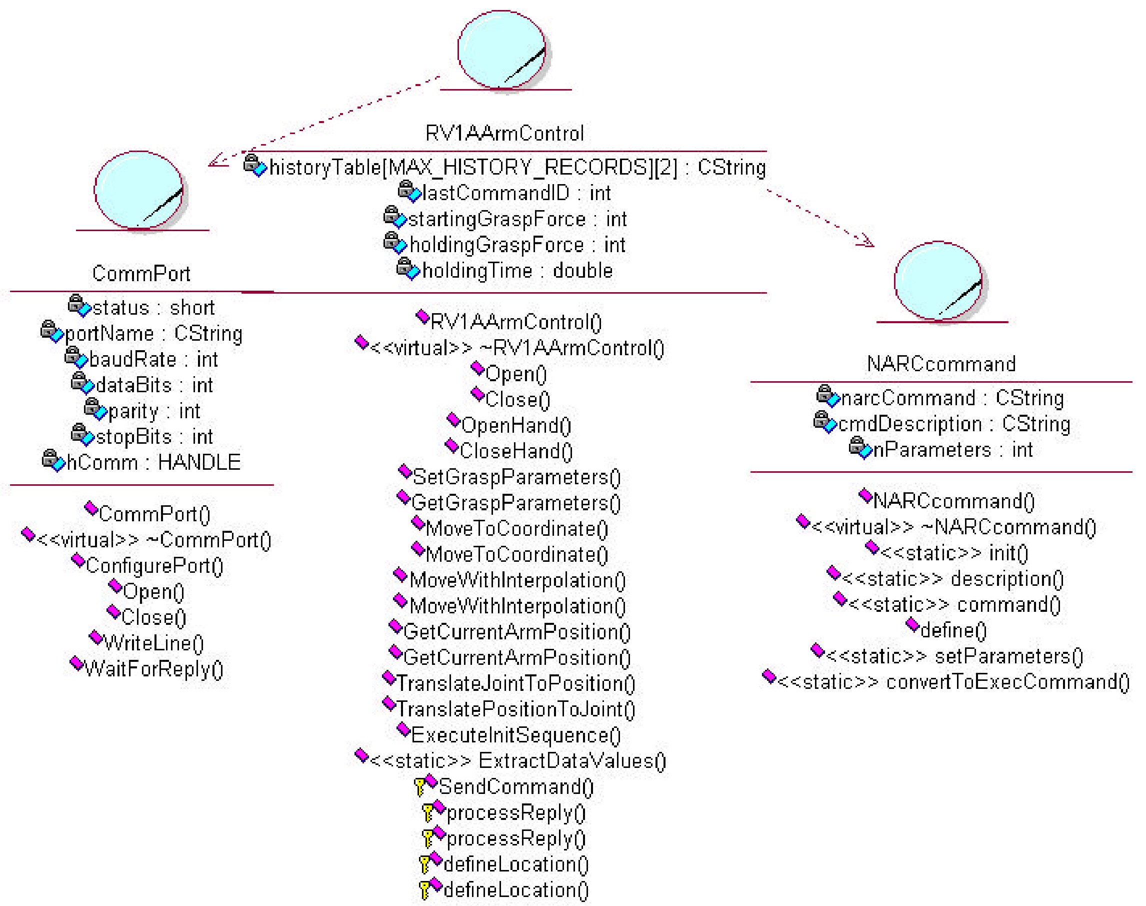


Figure A4.11: Robot classes design

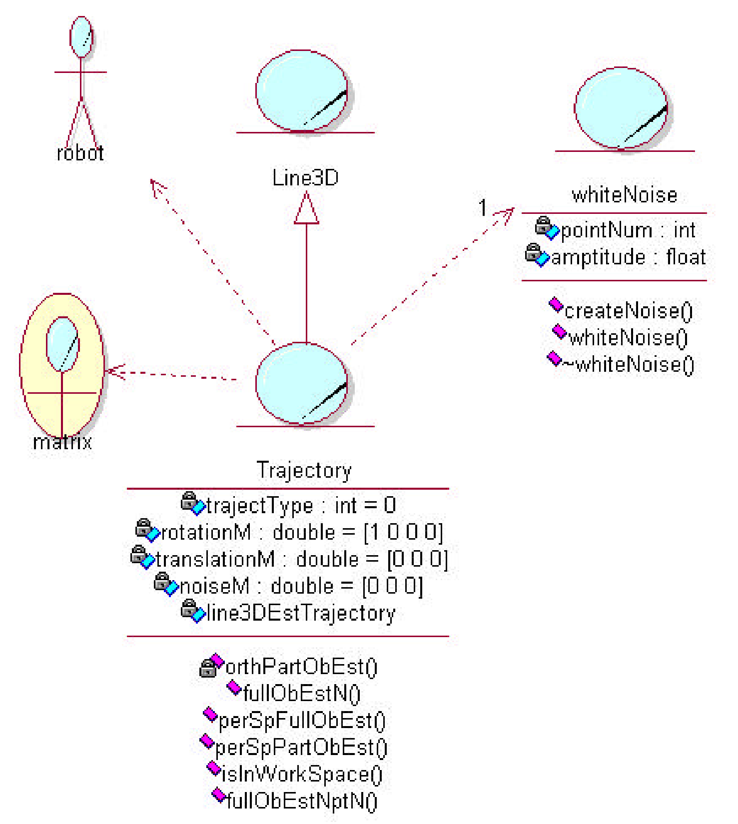


Figure A4.12: Trajectory classes design

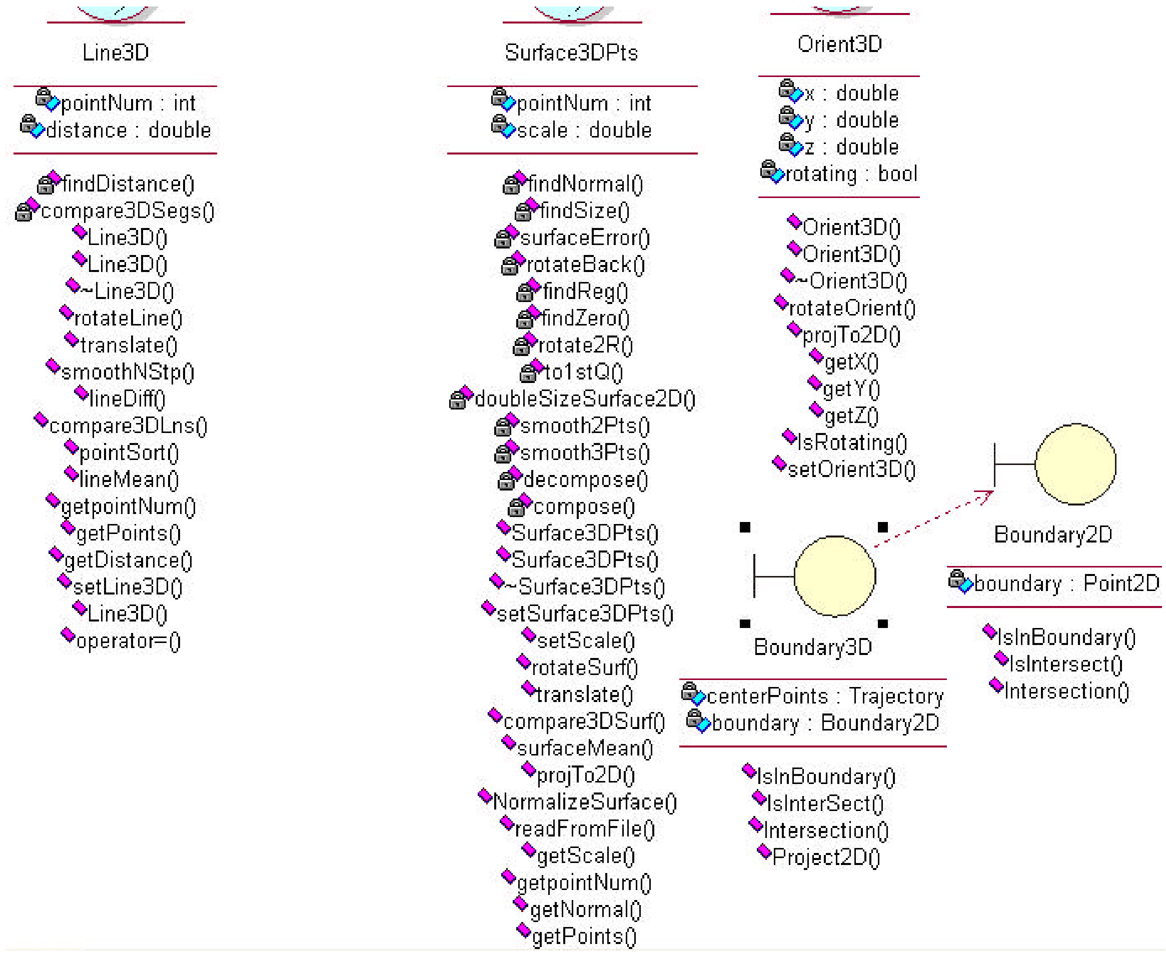


Figure A4.13: 3D Surface classes design

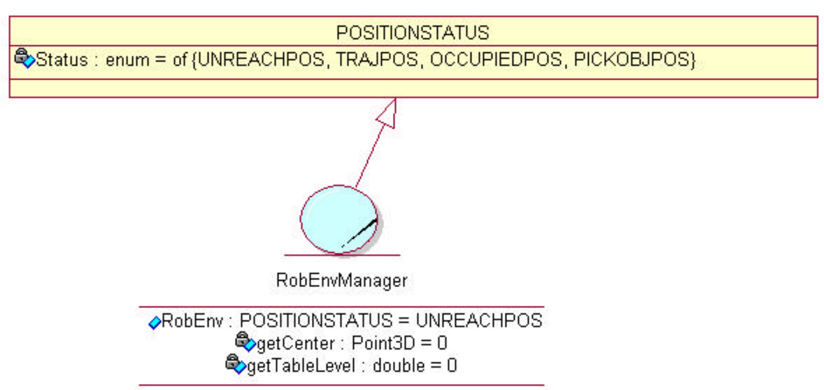


Figure A4.14: Robot manager classes design

Appendix V - REFES Final Report, Case Western Reserve

Milestone One: User Interface & Orientation of Phase I

Background and Motivation

FES systems for individuals with high tetraplegia.

Our interest in the REFES system is motivated by a significant ongoing effort by our research group to provide useful movement control to individuals with high cervical spinal cord injury, a condition referred to as high tetraplegia. These injuries are at the highest level of the spinal cord and leave those afflicted with extensive paralysis below the neck – typically such individuals are left with volitional control of just the head, neck, and in some cases shoulder shrug. Individuals with high tetraplegia are usually totally dependent on others for all aspects of care, and traditional rehabilitation procedures offer very limited options and result in limited functional improvement [Nathan and Ohry 1990; Lathem 1985].

Neuroprostheses are systems that apply controlled electrical stimulation to paralyzed nerves and muscles to restore function. These systems can be used to restore different functions to individuals with a variety of different neurological disorders, although many applications to date have been for individuals with spinal cord injuries. Specifically, neural prostheses based on functional electrical stimulation (FES) have been deployed for restoring upper extremity function [Peckham and Keith, 1992; Handa et al, 1992; Nathan, 1992; Prochazka et al, 1997], lower extremity function [Kobetic et al, 1997; Kralj et al, 1989; Triolo et al, 1996; Graupe 2002], bladder function [Creasey, 1996], and a number of other functions.

Specific to individuals with high tetraplegia resulting from high cervical spinal cord injury, several types of assistive devices can be used in conjunction with the retained movement function of the head and mouth to increase the independence. The mouthstick is the most widely prescribed piece of equipment for people with high tetraplegia [Lathem et al. 1985], and allows the user to engage in activities such as painting and computer use. However, the length of time one is able to engage in such activities is often limited by the strength and endurance of the neck muscles, and the use of a mouthstick is often socially compromising. Mobile arm supports such as balanced forearm orthoses (BFO) and ball bearing feeders are mechanical devices that support the arm and provide assistance to shoulder and elbow motions through a linkage of ball bearing joints. These devices are often abandoned following discharge from rehabilitation, however, because of poor functional outcome [Malick and Meyer 1978]. Environmental control systems can provide some independence under the control of eye gaze, tongue, rocking lever, or brow switch. However, such devices cannot perform essential personal tasks such as feeding or grooming. Robotic devices are also under development to enhance performance at the place of employment and at home [Hammel et al. 1989, 1992]. Users generally have a positive perception of the utility of the robotic assistant [Hammel et al. 1989, 1992]. However, these assistive devices are difficult to control and are not currently portable for use in the community.

The Cleveland FES Center is the world-wide leader in the development of functional electrical stimulation (FES) systems called “neuroprostheses” that substitute for the actions of the paralyzed nervous system by electrically activating the nervous system distal to the injury to elicit coordinated contractions of the paralyzed muscles that provide useful function. In addition to the work underway in the Cleveland FES Center to develop a neuroprosthesis for high tetraplegia, such systems have been

pursued by several groups in the past. Handa and his colleagues (Handa et al. 1987; Hoshimiya et l. 1989; Kameyama et al. 1990, 1991, 1993) have employed an FNS system to restore function in an individual with C4 tetraplegia. Their system attempts to restore movements by percutaneous stimulation of multiple muscles of the shoulder, elbow, wrist, and hand using stimulation patterns based on electromyographic (EMG) activity in able-bodied individuals. Pre-programmed sequences for different upper extremity activities are elicited by respiratory function (puff and sip). A balanced forearm orthosis was incorporated into the system to augment elbow flexion and shoulder stability and was identified as the most important factor in successfully utilizing their FNS system for functional tasks. Nathan (Nathan 1989; Nathan and Ohry 1990) has also reported restoration of function in high tetraplegia by FNS. This system used surface electrodes that were held in place by an elastic sleeve. Splinting and the use of a sling augmented voice controlled stimulation to the extremity. Two individuals with C4 level injuries have used the system to write and drink, and expressed the psychological benefits of seeing and feeling their arms move.

Figure 1 illustrates our general approach to a neuroprosthesis for high tetraplegia, along with our current idea of how the REFES system will be used as a component of this neuroprosthesis in the future. The implanted stimulators have already been developed and used in individuals with lower levels of spinal cord injury (and with less disability). The stimulating electrodes and the lead wires that connect them to the stimulator units have also been developed already and are in wide use in other systems. The REFES system potentially fills two critical roles in a neuroprosthesis for high tetraplegia that have been difficult to fill in the past: (1) a reasonably natural and effective command interface through which the user tells the neuroprosthesis what they want their arm and hand to do and (2) a sensor of arm position in 3D space that does not require the user to wear a large network of externally mounted sensors. The command interface is particularly critical because the neuroprosthesis for high tetraplegia requires the

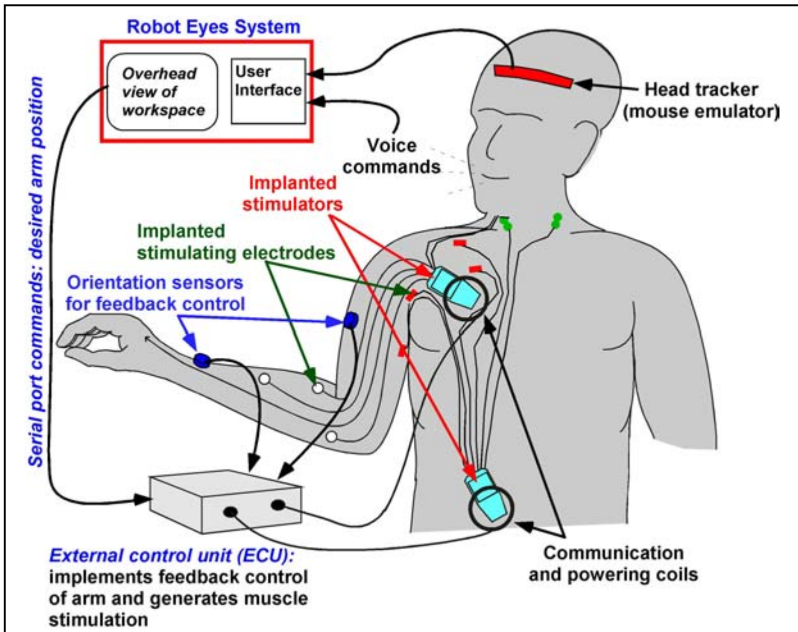


Figure 1: Schematic representation of REFES integrated in to neuroprosthesis for an individual with high level spinal cord injury. Two implanted stimulators produce needed contractions by passing current through the implanted electrodes into the paralyzed muscles (the "plant"). Orientation sensors are used to measure arm joint angles so that a feedback controller for arm position can be implemented. The external control unit provides power to the implanted stimulators via inductive links and provides the computational capacity needed to implement the feedback control algorithm. The user selects an object on the REFES display screen either by using a head pointer (a mouse emulator) or by voice commands. The REFES system determines which object was selected, computes a trajectory that moves the arm to the desired object while avoiding obstacles, and sends this trajectory via a serial port to the external control unit to provide the movement command.

development of appropriate control algorithms to control multiple degrees of freedom of the arm and hand in individuals who have very few voluntary movements that can be used for control. A number of approaches have been proposed (see discussion of Task b below), but all are difficult and tedious because the user must continuously command the position of the arm as

it moves to acquire an object and then somehow provide a separate command to open and close the hand around an object of interest. The primary advantage of the REFES system as a command interface for high tetraplegia is that the user need specify only the object he or she wishes to acquire via a simple visual interface. The REFES system then automatically computes a trajectory from the current location to the desired location that avoids any obstacles and approaches the desired object in an appropriate manner. For example, if the user specifies a coffee mug, the REFES system could program a trajectory that moves the hand to the mug handle. This approach could completely relieve the user of the burden of continuously controlling a trajectory through a cluttered workspace. The ability to recognize and acquire a wide range of items useful in everyday activities has the potential to significantly enhance the independence of the neuroprosthesis user and to decrease the amount of caregiver time needed each day (with substantial financial savings).

The Cleveland FES Center is currently deep in the development phase of the neuroprosthesis for high tetraplegia. No individuals with high tetraplegia have yet been provided with a neuroprosthesis, although our initial human implementation scheduled in a two-stage procedure over the next 12-18 months. In the absence of paralyzed subjects with neuroprostheses, our current approach to developing human command interfaces, including one based upon the REFES system, is to use a robotic simulator. Able-bodied subject controls a robotic arm with dimensions similar to the human arm just as if it were their own paralyzed arm, an effective simulation of an individual with high tetraplegia. Such an approach allows us to rigorously evaluate potential command interfaces BEFORE we actually implement such a neuroprosthesis in an invasive and expensive surgical procedure. The robotic simulator developed during this contract will be described below in the discussion of Task e.

Task a: Specify user/patient tasks

For individuals with high tetraplegia, our functional goals are to restore the ability to bring the hand to the mouth to allow feeding and grooming activities and to allow the arm to reach out to manipulate objects with the hand within a limited workspace in front of the body (e.g., to acquire food). These goals may appear to be modest, but the functional and psychological impact of providing even these simple functions to a group of individuals with essentially no upper extremity function cannot be overstated.

A user of a RobotEyes-aided neuroprosthetic system would be seated in a wheelchair and working on a horizontal workspace such as a lap board or a table. We have identified the following functions as being important goals for the neuroprosthesis for high tetraplegia:

1. Eating - the user would need to acquire a fork or spoon that may be placed either horizontally on the table or vertically in a holder. After acquiring the utensil, the user would need to bring it to a plate or bowl, acquire the food, and bring the utensil to his/her mouth. The user would then either acquire more food, or return the utensil to its initial location.

2. Drinking – the user would need to acquire a cup, either by grasping around the cup or by grabbing a handle. After acquiring the cup, the user would need to bring the cup to his/her mouth. The user would then return the cup to its initial location.
3. Telephoning (assume the user has a speakerphone) – the user would need to acquire a pencil or similar pointing device. The user would then move his/her hand to above the phone buttons. The user would push the buttons with the pointing device to initiate the phone call. After completing the phone call, the user would return the pointing device to its initial location.
4. Inserting/removing a computer disk – For a CD drive, the user would need to first move his/her hand in front of the CD drive and push against the button to open the tray. For CD and other disk drives, the user would need to go to where the disk is located (which could be vertical or horizontal) and acquire the disk. The user would then move in front of the disk drive and place the disk in the tray or slot, and push the disk or tray in to complete the insertion. To remove the disk, the user would need to move in front of the disk drive and push the button to eject the disk. The user would then need to acquire the disk and return it to its original location.
5. Wiping – the user would need to move over to a towel or tissue that is placed on the table. The user would need to acquire the towel, and then bring it to his/her face. The user would then move his/her head to allow sweat to be wiped off, to wipe his/her nose, or to scratch an itch. The towel or tissue would then be returned to its original location. Note that this mundane function is often mentioned by individuals with high tetraplegia as the function they would most like to have.
6. Brushing – the user would need to acquire a hairbrush, which may be placed either horizontally on the table or vertically in a holder. The user would then bring the brush to his/her head, and use a combination of arm and head movements to brush his/her hair. The user would then return the brush to its original location.
7. Reading – the user would need to move his/her arm to where the book or magazine is located, which could be either vertical or horizontal. The user would then need to acquire the reading material and place it either horizontally on the table or on a stand.

Task b: Study available interface methods and modalities

In general terms, the goal of this project is to determine the most appropriate method of enabling individuals with high tetraplegia to command the movement of their arm and hand through the measurement of body functions that remain under voluntary control. This objective is difficult to attain, however, because these individuals have very few voluntary movements that can be used for a command interface and because the number of motions that need to be restored to provide arm and hand function is considerable. Based on our past experience, we have identified criteria that should be satisfied by any acceptable command interface:

- Control of multiple motions (e.g., wrist, elbow, and shoulder) should be simultaneous rather than serial.
- The command method should be as natural and unobtrusive as possible
- The command method must not interfere with the patient's other abilities (talking, breathing, driving wheelchair).
-

As will be described throughout this report, the REFES -based interface that was developed in this contract satisfies the first of these criteria in a spectacular way that cannot be matched by any other existing user interface. This is because the REFES system operates at a higher level – the user specifies only the goal (e.g., to acquire an object), the REFES system computes the needed trajectory, and the neuroprosthesis drives the arm along this trajectory. The user is not bothered by controlling either multiple joint angles or multiple endpoint positions. However, the user must still somehow indicate the

targeted object to the REFES / neuroprosthesis system using a command interface that satisfies the second and third criteria. The following paragraphs will review a wide range of potential command interfaces

Individuals with high-level SCI are able to routinely operate wheelchairs, environmental control units and computers. In addition, control interfaces for rehabilitation robots have also been developed and tested. Popular control methods for wheelchairs and environmental control units are sip/puff input, head/chin switches and chin-controlled joysticks. Mouthsticks are used to directly operate switches, computer keyboards, for writing and for painting. Recently, voice recognition software has significantly improved, and voice control has become a common method for computer input and as a control for environmental control units. Another recently developed input device is the tongue-touch keypad, which allows the user to depress switches using his/her tongue (newAbilities Systems, Inc., Palo Alto, CA).

Control over robotic aids is provided though the use of an externally mounted device, such as a chin mounted joystick [Seamone and Schmeisser, 1985], a push button switch [Bach et al., 1990], or a head orientation sensor [Regalbuto, et al., 1992a,b]. In addition, voice recognition systems [Regalbuto, et al., 1992a,b] have also been examined as potential user interfaces because they do not depend upon external hardware. In all these cases the control over the robotic aid was accomplished using a menu-driven interface to select pre-programmed movements. In addition, some systems provide the possibility for free control over the robot movements [Hammel et al., 1989; Hammel et al., 1992], but this is accomplished serially (i.e. the individual first moves the 'shoulder' and sets the position, then moves the 'elbow' and sets the position, and so forth).

Similar methods of control have been used in functional neuromuscular stimulation (FNS) demonstration systems developed for high tetraplegia by our group and others. The surface stimulation system developed by Nathan [Nathan, 1984; Nathan and Ohry, 1990] used a voice recognition system as the control input. The interface was capable of recognizing twelve verbal commands that controlled system state, hand opening and closing, and movement of the hand towards and away from the face. Researchers in Sendai, Japan made use of sip-puff command input as well as voice control [Handa et al., 1985; Hoshimiya et al., 1989; Handa et al., 1992]. In our laboratory, we have used shoulder position control for those individuals who retain some activation of the trapezius muscle, and this has also been used successfully elsewhere [Betz, et al., 1992]. It should be noted that in all of these demonstration systems, shoulder support was provided through external bracing and, in some cases, other braces were used as well in order to minimize the degrees of freedom that must be controlled.

Type of Control	Applicable	Implantable	No Activity Interference	No Visual Interference	Parallel Input
Head Switch	+	+	+	+	o
Chin Joystick	+	o	+	+	+
Siff-Puff	o	o	o	+	o
Eye Gaze Direction	+	o	+	o	o
Voice	+	o	o	+	o
Tongue-touch Keypad	+	o	o	+	o
Shoulder Position	o	+	+	+	o
Head Orientation	+	+	+	+	+
Facial Myoelectric Signal	+	+	+	+	+
Electroencephalogram	+	+	+	+	o
Electro-oculogram	+	+	+	+	o

+ = positive quality
o = negative quality
Applicable: applicable to all injury levels (C4 and higher)
Implantable: amenable to implantation
No Activity Interference: does not interfere with the ability to talk or eat
No Visual Interference: does not interfere with ability to focus on hand
Parallel Input: allows multiple commands to generated simultaneously

Table 1: Potential high level SCI command methods

Table 1 summarizes the wide range of command methods that *could* be used in conjunction with the proposed REFES / neuroprosthesis system. Head switches are frequently already being used by the tetraplegic individual for a range of functions (e.g., wheelchair recline features), so we have chosen not to pursue this method. Chin operated joysticks, sip-puff inputs, and the tongue-touch keypad are appealing because they have all been used for a variety of control purposes in high tetraplegia, including environmental control, wheelchair control, and rehabilitation robot control. However, they all require the use of the mouth, precluding the execution of important functional goals related to eating and oral hygiene. Eye gaze control, in the form of the electro-oculogram, may be a useful command source because eye movements are very accurate and rapid, and can be performed in a cosmetically acceptable manner. However, eye movement based control will be useful only in situations where the user is not required to look at an object that is in a different position from the desired hand location. Shoulder position control will not be available to all users with high tetraplegia and provides limited information. Electromyographic recordings from muscles of the face are currently being examined in our group, but to be cosmetically acceptable this approach requires implantation of recording electrodes in the face. Furthermore, users are required to perform unusual facial expressions that may not be acceptable to them. We have also demonstrated the potential of the electroencephalogram as a neuroprosthetic control input [Lauer et al., 1999], but this form of control will require significant development in order to be applicable.

We have therefore narrowed our search of potential REFES user interfaces to head orientation and voice recognition. These input methods are attractive because they (1) are widely available, (2) are inexpensive, (3) are reasonably natural and unobtrusive, and (4) can be readily integrated into the REFES interface. Because of the short duration of this contract, it was decided to pursue commercially available input devices only. Bill Memberg, one of the engineering staff working on this project and an expert on assistive devices for disabled individuals performed an extensive survey of commercially available input methods (see full report in Appendix II). Based on this review, one voice recognition software package (QPointer Hands Free) and two head pointer devices (Madentec Tracker 2000 and Boost Technology Tracer) were selected as the interface devices to be evaluated. These are briefly reviewed in the following paragraphs.

- QPointer Hands Free

- This is a completely hands-free computer control by voice. Intuitive operation of any application by voice and full voice control over the Windows environment. Includes all capabilities of QPointer Keyboard.
- web site: http://www.commodio.com/products_voice.html
- purchase site: http://www.infogrip.com/product_view.asp?RecordNumber=741&sbcolor=%23CC9966&option=pointing&subcategory=12&CatTxt=Alternative&optiontxt=Pointing
- Manufacturer: Commodio
- Price: $189

Personal review (Bill Memberg): *Although Dragon Naturally Speaking and IBM Via Voice are more mainstream and better for continuous speech, the QPointer software might do exactly what we want. It takes a screen image and puts tags on objects on the screen, then lets you select the tags by voice input. It also works on all Windows programs, whereas I think the other voice engines only work on specific applications.*

- Tracker 2000

- Tracker 2000 allows you to smoothly move the cursor on the computer simply by moving your head, regardless of your disability. Tracker 2000 sits on top of the computer and tracks a tiny reflective "dot" worn on your forehead or glasses. When you move your head, Tracker 2000 elegantly converts that into computer mouse movements.
- web site: http://www.madentec.com/
- purchase site: http://www.infogrip.com/product_view.asp?RecordNumber=124&sbcolor=%23CC9966&option=pointing&subcategory=13&CatTxt=Head+Controlled&optiontxt=Pointing
- Manufacturer: Madentec
- Uses infrared
- Price: $1595

- Tracer

- Boost Technology's Tracer is a mouse that you control with your head. Tracer gives mouse control to people with Quadriplegia, Cerebral Palsy, Muscular Dystrophy, Multiple Sclerosis, ALS, Carpal Tunnel Syndrome and any other disability where the user lacks the hand control to use a standard mouse but retains good head movement. Tracer uses a small gyroscope to sense the user's motion. The gyroscope communicates wirelessly with the computer. Because it’s patented micro-gyroscope technology is remarkably precise - down to individual pixel resolution - anything that can be done with a mouse, you can do with Tracer. Draw. Surf. Design. Communicate. Connect.
- web site: http://www.boosttechnology.com/
- purchase site: http://www.infogrip.com/product_view.asp?RecordNumber=506&sbcolor=%23CC9966&option=pointing&subcategory=13&CatTxt=Head+Controlled&optiontxt=Pointing
- Manufacturer: Boost Technology

- Uses gyroscopes
- Price: $795

Personal review (Bill Memberg): *There are several options here, but based on reviews I have seen from assistive technology people, the Tracer and the Tracker One may be good options. The Tracer uses a gyroscopic device mounted on a visor or baseball cap, while the Tracker One (and its big brother Tracker 2000) uses infrared reflected off a dot stuck to the forehead. I would suggest getting both, so we could try out both approaches. They both use a USB port, which allows for easy setup (They both come in PS/2 versions too). The HeadMaster and HeadMouse also get good reviews, but may have a more difficult setup for our needs. The Smart-Nav AT is cheaper, but is less sensitive and has more problems with infrared 'noise'.*

Task c: Analyze tasks and interface methods and determine desirable matches

Based on the relatively simple restored motions that will be initially targeted for individuals with high tetraplegia and the review of available user input methods described above, we determined that a "workstation" solution would provide substantial benefits to the users while being feasible in the near future. Most of the tasks described above as most important to individuals with high tetraplegia involve objects located on a horizontal surface such as wheelchair lapboard or a tabletop. This is exactly consistent with the typical configuration of the REFES system. Furthermore, the large number of commercially available user input devices that could be used by the REFES system with little or no modification make a clinical realization within the next two years highly feasible. Limiting the targeted functions to the horizontal surface covered by the current implementation of the REFES system does limit the mobility of the user to a fixed site. However, most activities that will be performed by individuals with high tetraplegia are likely to be focused on a lapboard attached to their wheelchair or on a table in front of them. Furthermore, the user interface provided by the REFES system has the enormous advantage of requiring the user to specify only the overall movement goal (i.e., "pick up the cup") rather than requiring them to continuously control all aspects of all movements. We believe that this benefit far outweighs the limitations imposed by needed to operate in a fixed workstation environment.

In the future, REFES is expected to be reduced in size sufficiently to have the "workstation" attached to the wheelchair, eliminating this initial limitation. In this project we focused on user interfaces based upon the mouse port and voice recognition, but many other approaches can be accommodated via these methods. For example, ongoing work in the Cleveland FES Center is examining the use of facial EMG, eye movements, and brain recordings as natural and effective command sources. The use of these methods in neuroprosthesis users will require additional development of neuroprosthesis hardware and software, but the interface to REFES would need only trivial modifications.

In summary, the choice of a fixed workstation environment and a mouse/voice user interface do impose some limitations in the functions that can be restored. However, these limitations are minor and still allow for significant function to be restored. The chosen user interface methods integrate seamlessly into the REFES system because they are an intrinsic part of the Windows operating system.

Task d: Develop interface requirements

The intended long-term use of the REFES system in a neuroprosthesis was the primary factor in determining the interface requirements pursued in this project. These requirements are:

- A robotic simulator with dimensions and joint motions similar to the human arm will be used as a proxy for the paralyzed arm of an individual with high tetraplegia. This robotic arm can be controlled by able-bodied subject just as if it were their own paralyzed arm, an effective simulation of an individual with high tetraplegia. Such an approach allows us to rigorously evaluate potential command interfaces, such as the REFES system, BEFORE we actually implement a neuroprosthesis for high tetraplegia in human user, deferring an invasive and expensive surgical procedure until we have high confidence that the neuroprosthesis will be successful.
- The REFES system must work through the same controller hardware used by the rest of the neuroprosthesis. In particular, the REFES system must be compatible with a high-performance controller based on single board computers that is currently in the prototype stage in our laboratory.
- Likewise, the interface between the REFES system and the neuroprosthesis must be implemented in the next generation neuroprosthesis software development tools under development in our laboratory. Specifically, this means that the REFES -to-neuroprosthesis interface must be implemented in Simulink (The Mathworks, Inc.) and executable under their "xPC Target" real-time environment.
- After discussion with SIS, the following interface specifications were agreed upon:
    - The REFES -to-neuroprosthesis communications interface would be a serial port because this could be easily implemented on the REFES system and was accessible to the Simulink/xPC Target environment.
    - The user of the REFES /neuroprosthesis system will be provided with a top (overhead) view of the workspace in front of them. Objects within the scene will be indicated in their proper locations. These objects are to be listed in an on-screen menu or each object could be labeled at its location on the screen. The cursor is then placed over the label and the object selected using the mouse emulator.
    - The user will specify only the desired endpoint of a movement and the REFES system will compute the needed trajectory, taking into account any obstacles. This required the development of the inverse kinematics for the Staubli robot used at CWRU, which is different than the robot used by SIS.
    - The two basic input methods described above (head tracker mouse emulator and voice recognition) were selected for study within this contract.

Task e: User interface hardware development

Figure 2 illustrates the overall system that was set up during this contract to interface the REFES system into a realistic neuroprosthesis simulator and then evaluate the REFES system as an effective neuroprosthesis command interface. The neuroprosthesis user (the "Operator" in Figure 2)

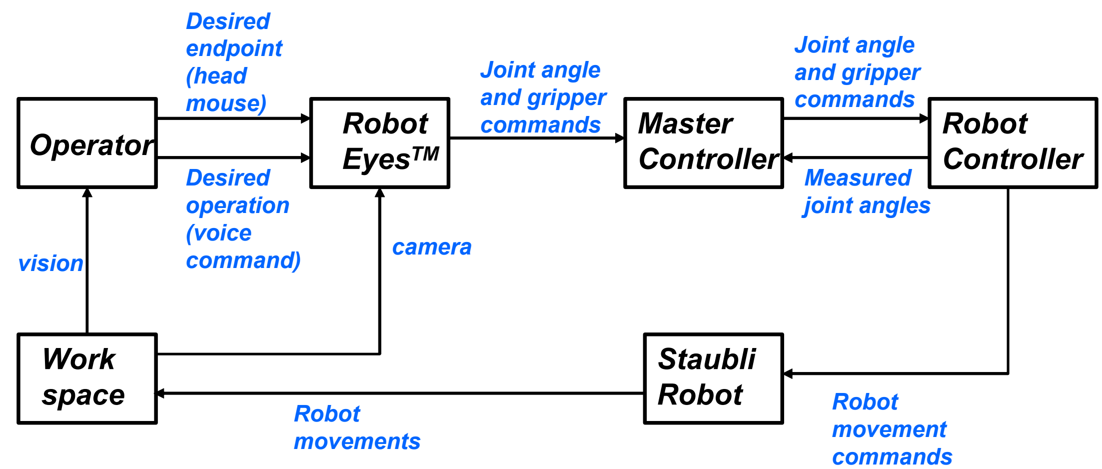


Figure 2: Conceptual block diagram illustrating the integration of the REFES system into our user interface evaluation system

provides commands to the REFES system through some combination of mouse commands generated by the head tracker and the voice recognition system. These commands specify an object in the scene that is to be acquired. The REFES system then computes a trajectory that will approach the object from an appropriate angle while avoiding other objects in the workspace and any other obstacles. These trajectory commands are transmitted to our xPC Target-based Master Controller for execution. In this contract we used a robotic simulator rather than an actual human subject, so the Master Controller passed the REFES trajectory commands through another serial port to the controller for the Staubli robot used. In a real neuroprosthesis, the commands would instead be routed to a stimulation system that activated the appropriate set of paralyzed muscles in a temporal pattern that will drive the human arm to the desired target. The user and the REFES system observe the movement of the object to its new desired location, preparing both to make another movement if needed.

The following paragraphs will describe the major components of this system in detail.

User input interface to REFES system

As described above, the selected head tracker mouse emulation systems interface directly with standard computer mouse ports, so no hardware development was necessary.

The QPointer voice recognition software is an add-on to Windows, but it requires no special software development because uses the built-in voice recognition features of the operating system. QPointer works by identifying text tags of features on the desktop and any open windows. Various types of text can be displayed, including icon labels, button labels, and window names. Recognized text is indicated by a pop-up number next to the text. Multiple instances of the same word are sequentially numbered, starting at zero. Speaking the number of the text you wish to select places the cursor at that point and left-clicks on the text. REFES uses this feature by having all items in the interface identified with unique text tags. QPointer is used to activates the various REFES functions simply be speaking the name of the button or object to be manipulated and then its number

- Robot simulator for human arm

We currently have an installed robot, mounted inverted and at an appropriate height to approximate a human arm. The Staubli RX60 robot was selected because its maximum reach of 660mm is almost identical to the average human reach of 650mm. The maximum payload of this robot is 4kg at low speed and 2kg at full speed, both more than adequate for simulating the arm function of individuals with high tetraplegia who are expected to be relatively weak even with a high performance neuroprosthesis. Because of the relatively large dimensions and mass (44 kg) of this robot and its

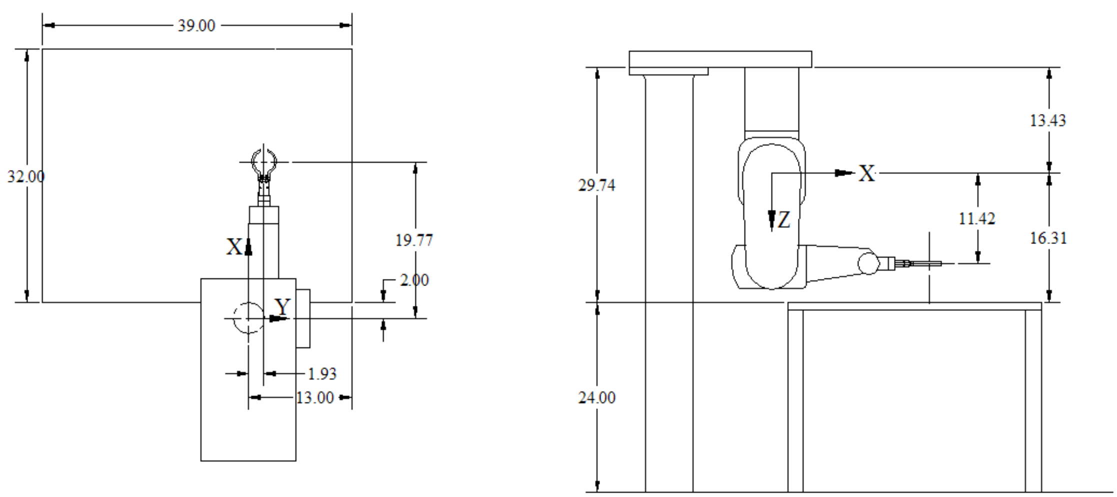


Figure 3: Mechanical drawing used to construct the inverted mounting fixture for the Staubli robot.

unusual inverted mounting arrangement to simulate a human arm, a substantial mounting fixture was required. Figure 3 illustrates the mechanical drawings used to fabricate this mount. Essentially, this is a large steel tube with mounting flanges at either end. The lower end was bolted to the floor using eight concrete anchors. A large aluminum block was bolted to the other end, with the robot cantilevered off the end of this block and hanging downwards. Photographs of the robot mounting arrangement are provided in Figures 4 and 5.

The nominal horizontal workspace provided by the robot mounted in this configuration is illustrated in Figure 6. The area of this workspace is just slightly larger than would be expected to be accessible to an individual with high tetraplegia with an advanced neuroprosthesis that restores arm motions.

(a) (b)

Figure 4: Photographs of the Staubli robot arm mounting in starting position (a) and a typical application position (b). A meter stick is shown in part (a) as a reference.

Figure 5: Photograph of the Staubli robot with pneumatic gripper mounted at the endpoint, in a functionally relevant position.

The Staubli robot includes a controller that can be used in the typical manner for an industrial robot. Because our goal is to emulate a person with spinal cord injury using a neuroprosthesis and because we need to interface to the REFES system, we have implemented a PC-based Master Controller (MC) that is capable of very fast real-time control and provides a wide range of interfaces. It receives the robot joint angle trajectories from the REFES system, manipulates them as needed to emulate a paralyzed arm, and then send the needed commands to the Staubli robot. These joint angle commands are sent to the robot via a serial interface (115200 bps, 8N1). The joint angles are updated at 100 Hz for smooth operation. A checksum error checking routine has been implemented to account for dropped bytes in the data stream. Qualitative assessment of the communication between the MC and the robot was performed using transmission of simultaneous sinusoidal inputs to the joints and the arm tracks well.

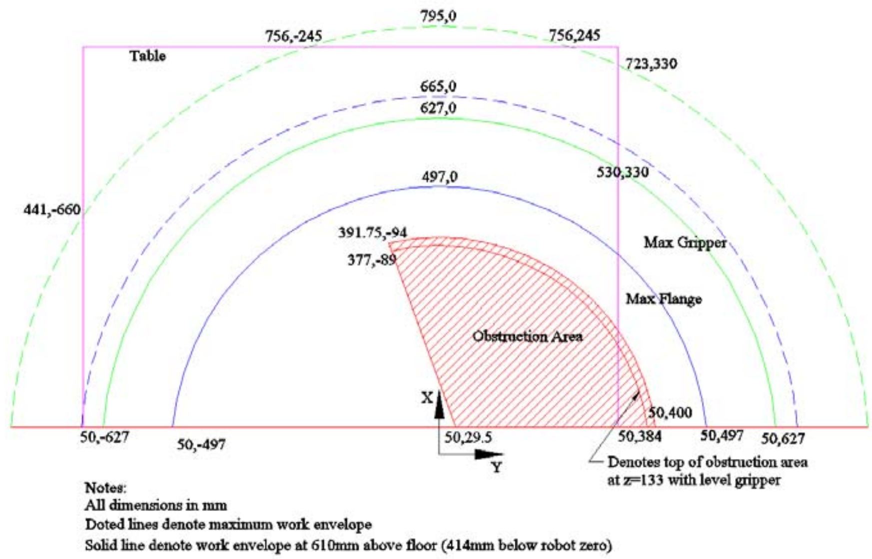


Figure 6: Nominal workspace of the Staubli robot as mounted. The table used in the various tests performed in this contract is indicated by the square purple box.

The kinematic equations for the robot arm were derived such that for a given wrist location and orientation, the requisite 6 joint angles are calculated. However, these equations are not currently used in the Master Controller as REFES system provides joint angles directly. Note that we derived the inverse and forward kinematic equations for the Staubli robot out of necessity, since Staubli considered this code proprietary and would not provide it to us. The derivation of these equations and the code used to implement the equations is contained in Appendix I attached to the end of this report.

REFES to robot interface

In the particular setup used in this contract, the REFES system by itself is more than capable of taking the user inputs, computing the desired trajectory, and sending the needed commands to our robot. We inserted an “extra” controller in this loop, however, so that the work performed under this contract would be hardware and software compatible with the other aspects of neuroprosthesis development that are continuing in parallel.

Neuroprostheses typically include electronic circuitry to generate appropriate stimulus pulses, electrodes to deliver the stimulation to paralyzed neural structures, a command interface through which the user indicates his or her intentions to the neuroprosthesis, and a control algorithm that acts to insure that user intentions are produced via the stimulated contractions. Clinically deployed neuroprostheses typically include highly customized command and control hardware and software that implement the simplest possible algorithms to minimize the size and power requirements of an “external control unit” (ECU) that either generates stimulus current pulses directly or sends commands to an implanted stimulator via a radio frequency link [Smith et al, 1987, 1998]. The low power microcontrollers typically used in the ECUs have limited computational power, so algorithms are often implemented as look-up tables rather than equations and are implemented using low-level “embedded” programming techniques. The use of sensors is typically limited to single command sources (e.g., a shoulder transducer) or simple event feedback (e.g., foot contact with the ground).

Many of these neuroprostheses have been very effective and robust. However, this approach does not scale well to accommodate significant improvements in a given neuroprosthesis (e.g., including a feedback controller as is needed for high tetraplegia) or to facilitate the development of more sophisticated neuroprostheses (e.g., restoring function at additional joints such as those required in high tetraplegia). The low power microcontrollers used do not have sufficient processing power to perform any significant signal processing, precluding the use of many potential command sources that require frequency domain processing, pattern recognition, or other straightforward but more computationally demanding techniques. This limited processing power, coupled with awkward, one-of-a-kind sensor interfaces, has also resulted in the virtual absence of modern control systems engineering techniques applied to neural prostheses. Finally, the need to produce highly optimized software code has required handcrafted, low-level assembly language approaches that have long development cycle times and are not readily accessible to anyone other than expert software engineers.

Figure 7: Versalogic single-board computer within prototype housing.

To address these limitations and make a neuroprosthesis for high tetraplegia feasible, we have developed a new Master Controller (MC) based upon a commercially available single board computer with greatly enhanced processing power and standard input-output interfaces for sensors. Specifically, the Versalogic VSBC-6 single board computer was used as the computational engine. This 14.6 x 20.3 cm (5.75” x 8.00”) printed circuit board, referred to hereafter as the VSBC, is based upon a 266 MHz Intel Tillamook processor and also features digital input-output lines, analog-to-digital converter input lines, 4 serial ports, an Ethernet port, and many other interfaces. In the current application (labeled “xPC Target” in Figure 2), the Master Controller reads the Robot Eye robot movement commands via one serial port and outputs them to the Staubli robot. As noted above, this Master Controller was included to insure that future neuroprostheses can interface seamlessly with the REFES system. A photograph of the Versalogic unit is shown in Figure 7.

Task f: User interface software development

<u>REFES user interface software development</u>. The software development for working with the REFES user interface was trivial because of design decisions made early in the project. The head tracker mouse emulation system simply replaced the standard mouse and no software development was needed. To use this device, the user simply moves their head side-to-side and nods up and down to move the cursor on the screen. As noted above, the QPointer voice recognition system utilizes the built-in voice recognition facilities of the Windows operating system and therefore required minimal modifications, and any modifications were implemented by SIS into the REFES system. In the final “functional” tests of the REFES system, the robot was commanded by a combination of head movements and voice commands: the head tracker mouse emulator was used to move the mouse cursor to the desired object. Once the appropriate object was highlighted, the QPointer voice recognition system was used to provide the equivalent of a mouse click by speaking the word “click”.

Real-time control neuroprosthesis software development

As noted above under Task d, we specified that the REFES system should work in a transparent manner with the software environment of the neuroprosthesis. After extensive internal discussions and consultation with SIS, it was determined to interface the REFES system to the neuroprosthesis controller via the next generation of real-time controller development tools from the Mathworks, Inc. Simulink is a block diagram-based programming and simulation method that is extremely intuitive and powerful. In addition to providing a highly intuitive programming interface, Simulink also provides access to a huge library of existing control systems, signal processing, and other relevant software toolboxes. It is also the programming interface to "xPC Target", a real-time operating system that provides high performance using standard Windows-compatible computers (in this case the Versalogic VSBC that serves as our Master Controller) as computational engines. The xPC Target Toolbox includes blocks specifically written for standard PC input-output ports (e.g., serial, parallel) and a wide range of input-output devices (e.g., analog-to-digital and digital-to-analog converters) from various manufacturers. Although not all Simulink blocks can be executed under xPC Target, the breadth and depth of functions that are compatible are quite large and certainly sufficient for the type of neuroprosthetic controllers currently conceived, including the serial interface to the REFES system developed here.

To establish this approach as adequate for any conceivable neuroprosthesis application involving the REFES interface, we performed a basic evaluation of the overall real-time performance of this single board computer system. Since Pentium processors perform one floating point mathematical operation (FLOPs or "floating point operations" per second) per clock cycle, an upper limit on the number of operations that can be performed during an inter-pulse interval can be obtained by dividing the processor clock frequency by the desired controller sampling frequency. For the Versalogic computer used here, this theoretical maximum is greater than 22 million FLOPs (i.e., the 266 MHz clock frequency divided by the 12 Hz stimulus frequency typically used in a neuroprosthesis). This calculation is not directly useful, however, because the number of FLOPs required by a controller algorithm depends upon software implementation details and internal machine latencies, whether these decisions are made by a human programmer or by the Simulink/xPC Target environment. To provide a more direct indication of the capacity for controller capacity, the Mathworks Inc. standard benchmark for xPC Target "xpcbench" was used to determine the number of continuous states (derivatives, integrators, etc.) that can be included in a real-time control algorithm for a specific processor. This benchmark determines the maximum sampling rate that can be supported by the specific hardware used for control algorithms of increasing complexity. The simplest algorithm consists of three simple blocks and essentially provides information on the non-computational overhead of the hardware. The remaining four algorithms all implement a simulation of a flight controller for the longitudinal motion of a Grumman Aerospace F-14 fighter plane, which consists of 62 Simulink blocks and 10 continuous states. The algorithms implement 1, 5, 10, and 25 of these F14 simulations (i.e., with 62 Simulink blocks and 10 continuous states, 310 Simulink blocks with 50 continuous states, 620 Simulink blocks with 100 continuous states, and 1550 Simulink blocks with 500 continuous states, respectively). The "xpcbench" was performed using the VSBC-6 with a Tillamook 266 MHz processor. The results from this benchmark indicate that controllers with at least 27,000 continuous states could be realized with the Versalogic single board computer. This far exceeds the complexity of any current neuroprosthesis design and indicates an enormous capacity for expansion for future neural prostheses.

In summary, the xPC Target real-time operating system, in conjunction with the associated Simulink programming environment, is capable of providing the serial interface needed to communicate with the REFES system. Extremely complex control algorithms can be implemented with ease using the Simulink interface, providing ample capacity for any neuroprosthesis system that will be coupled to the REFES interface. At a minimum, this approach will be used in the future to implement a feedback controller for the endpoint position of human users of neuroprostheses, perhaps using position signals provided by the REFES system (see Demonstration Task II, Task a).

Task g: Integration with REFES

As described above, the REFES system communicated with our Master Controller via a standard RS-232 serial port. The following section describes the communications protocol that was used.

- REFES Communication

The REFES (RE) system communicates with the Master Controller (MC) via single character ASCII commands corresponding to the various functions of the system. Data transmission is also in ASCII, following the format:

B1	B2	B3	B4	B5	B6	B7	B8
+/-	A	B	C	·	D	E	F.

These eight bytes indicate the sign, the value (up to 999.999), and decimal location to 1/1000 precision. This data structure is used between the RE and MC programs for both joint angles (commands and feedback) and spatial position feedback. These ASCII strings are then converted into decimal values for processing. A reply is sent back to the RE system following each command. In the case of joint angle commands, the reply occurs at the end of the joint data string. Unique replies are sent upon completion of path execution, and after receiving a request for robot position information. Table 2 contains a list of all commands used between the RE and MC systems.

Command				
Name	**Letter**	**ASCII**	**Action**	**Reply**
Start Up	S	83	Activates robot - not used	R
Shut Down	D	68	Shuts down robot - not used	R
Clear Path	P	80	Clears stored joint angle queue on MC	R
Execute Path	E	69	Runs the stored joint angle queue	R & L
Pause	A	65	Pauses robot motion	R
Resume	U	85	Resumes robot motion	R
Stop	Z	90	Stops robot motion and clears joint angle queue	R & L
Path Begin	B	66	Indicates beginning of joint angle queue	R
Path End	F	70	Indicates end of joint angle queue	R
Home	H	72	Move to home position	R
Move	M	77	Load single set of 6 joint angles	R
Open	O	79	Open gripper	R
Close	C	67	Close gripper	R
Get Joint Angles	G	71	Request for current robot joint angles	G

Get Spatial Location	X	88	Request for current robot spatial position	X

Table 2: List of commands sent to MC from RE and their replies.

Data communication between the MC and Robot uses a smaller structure to minimize the required bandwidth. Joint angle commands and feedback from the robot follows this format:

B1	B2	B3
Hundreds	Tens	Decimal

The decimal portion of the data is passed as a whole number and then converted back into a decimal with 1/100 precision. To avoid negative numbers, all angle data has 180 added to it, while spatial position information is increased by 665 and then divided by two. All data signals sent to the robot are passed through a saturation block with a cut-off at 255 to eliminate crashing receiving the serial port.

Milestone Two: Preliminary FES Requirements & Planning

Task a: Identification of FES range of motion

The range of motion that the ultimate REFES / neuroprosthesis system will achieve will undoubtedly be limited primarily by the FES system, not by the REFES system. Because of muscle weakness due to disuse atrophy and denervation, it is highly unlikely that muscle strength will reach more than 25-50% of able-bodied levels, limiting the range of possible arm movements to shoulder level and below. Since these individuals will also lack voluntary control of muscles of the torso, the radius of possible motions will be limited to the length of the user's arm. Finally, individuals with high tetraplegia will almost always perform functional tasks from the base of a lapboard or a tabletop. Thus, the likely range of motion that we can restore will be from approximately the lower abdomen to shoulder height, with an outreach reach limited to the length of the arm. The functional tasks targeted by our neuroprosthesis reflect this expected workspace and are focused on restoring the ability to bring the hand to the mouth to allow feeding and grooming activities and to allow the arm to reach out to manipulate objects with the hand within the limited workspace in front of the body (e.g., to acquire food).The workspace provided by the Staubli robot used in this study closely replicates this limited workspace.

Task b: Identification of RE-FES command parameters

The command structure of the communication between the REFES system and the Master Controller was described in detail above (Milestone I, Task g). Briefly, the REFES system computes the trajectory of joint angles needed for move the endpoint of the arm from the current position to that of the specified object. These joint angle commands are then transported via the serial port to the Master Controller, which then passes them on via a second serial port to the Staubli robot for actuation.

- System Start-Up

The following steps are required to get the complete three-component system ready for operation. The order listed is advised for best performance.

Robot

1) Turn on main power
2) Turn on air compressor
3) Once powered up, select "Application Manager"
4) Open the program "fullsrl_jtNpos2" from the list of files on the disk

Master Controller

1) Launch Matlab and change the operating directory to the location where the program is stored
2) Load "RS232Async_send" structure to workspace
3) Open the model "REreader_sender6"
4) Build model to compile to xPC Target

MC/Robot System

1) Run the program "fullsrl_jtNpos2" on the robot
2) Type "+tg" at the Matlab prompt to start the MC (must be completed within 1 minute of starting the robot program)

REFES

1) Launch the RE software from the icon on the Desktop (MC must be running for RE to start up)
2) Press the "Capture Data" button on the RE interface

Task c: Begin construction of FES simulation

The interface that we have chosen between the REFES system and the neuroprosthesis for high tetraplegia has greatly simplified the "FES simulator" that is needed to evaluate this interface. As noted above, the available workspace used in this contract is very similar to that likely to be available to the user of a high tetraplegia neuroprosthesis. We also considered including other limitations to the neuroprosthesis/Staubli robot system to make it behave more like a paralyzed arm under the control of a neuroprosthesis. In particular, we considered limiting the speed of the Staubli robot (to reflect inefficiencies in the command interface and abnormal muscle properties) and/or adding random positional noise (to reflect limitations in the ability of the user to produce steady commands). Both of these limitations would be extremely simple to implement. After much consideration, however, we decided that these limitations were artificial given the overall expected configuration of the REFES /neuroprosthesis system. First, users of this system have indicated to us that speed is NOT an important factor – they do not care if the movement is one-half or one-third as fast as the comparable able-bodied movement as long as it can be successfully executed. Furthermore, the existing interface to the Staubli robot has not been optimized and limits movement speed to a range that would be expected in a real neuroprosthesis, so no further speed limitations were deemed appropriate. Second, the overall command interface provided by the REFES /neuroprosthesis system has unique properties that make it inherently insensitive to the poor properties of the paralyzed arm. Specifically:

1. The user does not have to provide continuous control of the desired location of their arm. The user provides only the final goal of the movement, with REFES doing the difficult task of selecting an appropriate trajectory and the neuroprosthesis feedback controller insuring that this trajectory is indeed followed. Other command methods will require significantly greater attention by the user, with the real danger of cognitive fatigue. Furthermore, the user may not always be able to see the relative position of their hand and objects because the arm obscures vision in some locations. Finally, the user may want or need to look away from their hands in order to complete various tasks and could lose control via many other command interfaces. REFES provides a very elegant solution to all of these problems.

2. The feedback controller will automatically compensate for weakness and fatigue up to the maximum capacities of the muscles. No command interface can do better than this.

3. The REFES imaging system will exhibit much greater positional accuracy than we can hope to achieve via most other command interfaces.

In summary, we believe that the overall combination of the REFES system, the neuroprosthesis Master Controller, and the Staubli robot provides an environment that is relevant for simulating the use of different command interfaces for a high tetraplegia neuroprosthesis.

Task d: Begin construction of translation driver

The communication between the REFES system and the Master Controller of the neuroprosthesis was described under Milestone I, Task g. The software environment of the Master Controller was described under Milestone I, Task f. The REFES side of the communications scheme was implemented by SIS and will not be described here. This section will detail the Simulink programming implemented on the Master Controller to read the REFES commands and relay them to the Staubli Robot controller.

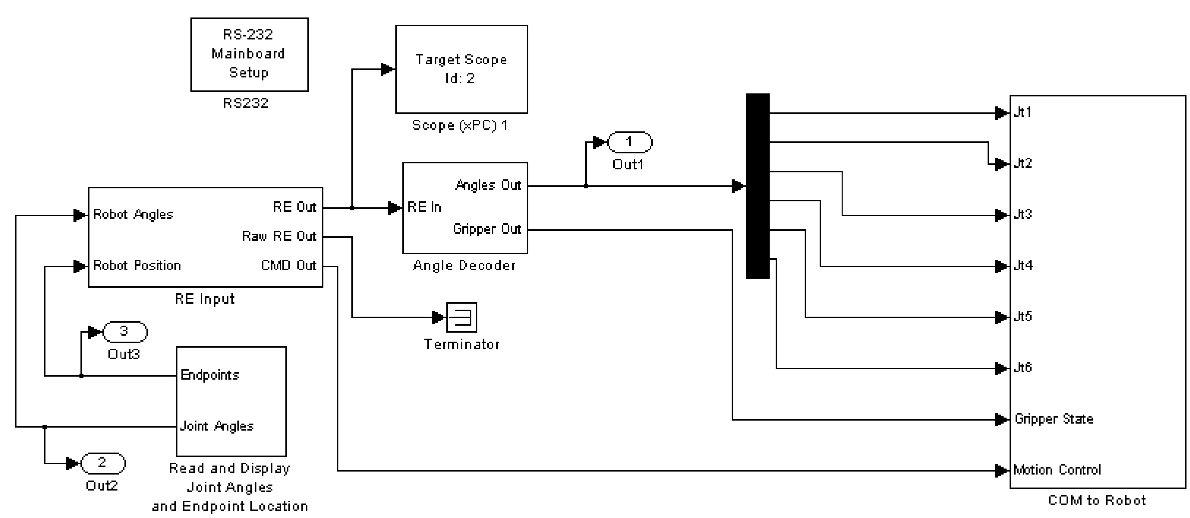


Figure 8: Main Simulink block that implements the “neuroprosthesis” software on the Master Controller.

Figure 8 illustrates the main Simulink block that implements the "neuroprosthesis" software on the Master Controller. The graphical nature of this block diagram interface makes this approach very intuitive and desirable. Note that each block within Figure 8 is really a separate "program" that can be expanded by double-clicking on the block to review another block diagram. The following sub-sections will describe the basic functions of the customized blocks illustrated in Figure 8. Appendix III includes the block diagrams for all of the Simulink blocks used by these blocks and sub-blocks.

Main block: Master Controller

The program for the Master Controller is created in Simulink using a series of graphical blocks and subroutines. These subroutines can be thought of as individual functions. This model is then compiled and downloaded to the xPC Target. Start and stop commands are given at the Matlab workspace prompt. The overall logic of the Master Controller can be summarized as reading in information from both the robot and RE system, converting into the appropriate format and passing it along. This logic structure can be seen in the top-most routine, with additional blocks for exporting data to the Matlab workspace and displaying the angles passed to the robot on the xPC Target monitor. A detailed description of the major components of this structure is below.

Read and Display Joint Angles

This subroutine contains the blocks to read the serial port (COM1), reconstruct the transmitted information into decimal values, and display on the monitor. The Reconstruct Angles block first locates the synchronization byte of the 37 byte string of data sent from the robot and then reorganizes it into the correct order (Byte Arranger). The data stream is then split and each half is reassembled into decimal vectors for joint angles and spatial position and orientation (Build Angles and Build Endpoint respectively). A series of blocks is also present to detect and correct for any noise errors.

RE Input

The RE Input subroutine is where the data is received form the RE system at the serial port (COM2), converted and executed as a command (Interpret Command), and robot commands are stored for later execution (Queue). Replies back to the RE system are also handled in this block, returning joint angles, spatial position, or path complete commands.

The Interpret Command block first determines which command was sent (Command Selector) and then replies that it received a recognized command (Serial Reply). The Load Position block formats the data from RE and places it into the queue. This includes commands for Home, and opening and closing the gripper. The Execute Path block pushes the stored joint positions (and commands if applicable) out from the queue to the robot upon an execute command from REFES . Pause, Resume, and Stop commands are also handled by this block.

Angle Decoder

This block translates the open and close commands from a vector of stored joint angles into a separate state command which is passed on to the robot on a separate signal line. A memory loop holds the angle output line at the last signal that was not an open or close command.

COM to Robot

The COM to Robot block splits the incoming joint angles into 3 byte pieces and passes them as a string, along with a synchronization byte, a checksum byte for error correction, the gripper state and pause/resume/stop commands. This 22 byte data string is passed through a saturation block to avoid crashing the serial port on the robot.

Design Review (tasks 1 through d)

Several key members of the CWRU team (Kirsch, Williams, Uhlir, Tyler) visited Spatial Integrated Systems on April 9, 2003. During this visit, many specific design decisions were agreed upon, including the nature of the REFES -to-neuroprosthesis interface, the user-to-REFES interface. See Task d for a description of these decisions.

Milestone Three: Demonstration Test 1

These tests were performed by Ardiem in conjunction with SIS and CWRU in December 2003. All of these tests were completed as proposed and the results will be provided by Ardiem.

Milestone Four: Demonstration Test II

Task a: Demonstrate feed back-controlled arm.

Much of the work for this task was completed at CWRU in collaboration with Ardiem and SIS in December 2003. The data from these tests were retained by Ardiem and SIS, so we cannot comment on them in detail. Our report will focus on the interpretation of these results in terms of neuroprosthesis control.

Ardiem performed a series of tests that examined the ability of the REFES system to provide continuous measurements of arm position in addition to the location of static objects within the workspace. If such measurements could be obtained accurately at a sufficient rate (> 20 Hz for neuroprosthesis applications), they could replace the body-worn sensors currently under consideration for closing a feedback loop for controlling arm position in the high tetraplegia neuroprosthesis (see Figure 1: orientation sensors). There are a number of potentially significant benefits that could be derived from such an arrangement:

- The user would not be required to wear sensors on their arm. This is a major advantage for a number of reasons. The costs of the orientation sensors are avoided. The neuroprosthesis does not have to provide power to the sensors. More importantly, the user does not need to put the sensors on each time the system is used and the sensor and its associated external cabling are eliminated, completely avoiding the vexing problem of interference with the very motions that are intended to be restored.
- Artifact in the body-mounting orientation sensor signals related to soft tissue motion (i.e., relative motion between the underlying bone and the sensor due to muscle, fat, and skin properties) will be completely avoided.
- The positional accuracy that is obtained from the REFES system is expected to be significantly better than that obtained from the orientation sensors (which require an accurate transformation matrix and highly repeatable location of the sensors across each use by the user or their caregiver).
- The REFES position tracking system has the potential to track objects other than the arm that may move in the workspace (e.g., someone else moves an object or the user

inadvertently bumps and moves an object). Being able to track such movements will preserve the obstacle avoidance capability of the REFES system, with the significant advantages described previously.

The tests performed by Ardiem on the feedback capability of the REFES system demonstrated that this is potentially feasible. In particular, the achieved frame rate of >20 Hz is more that sufficient for use in a neuroprosthesis system because the typical stimulation frequency (the fastest that any desired outputs can be changed) is 12-16 Hz. These results are very promising, and likely future enhancements of the REFES system (even faster updates, miniaturization) are likely to make this approach even more feasible.

Before a REFES -based feedback system can be safely implemented, however, several issues beyond the scope of this project need to be resolved. In particular, the following issues must be resolved:

- **The system must be able to accurately locate the endpoint location of the hand for different hand orientations.** The REFES system estimates mean endpoint position based upon skeletal landmarks on the hand, but the shape of the hand will change significantly when the hand is opened or closed, as well as when the orientation of the hand changes. The shape of the hand will obviously change as it opens around an object and then closes to grasp it. Furthermore, the orientation of the hand needed to acquire different objects will vary depending on the shape and orientation of the object. To be incorporated into a neuroprosthesis system, the REFES system must be able to provide a robust measurement of endpoint location regardless of hand orientation or whether the hand is open or closed. This functionality could be accomplished by imaging additional features (different skeletal landmarks, differences in color, other locations on the arm) that are less sensitive to hand orientation. Alternatively, the user could wear several objects (e.g., rings) that present a high contrast to the imaging system and have distinctive shapes that allow robust identification of their orientation. This approach is somewhat less desirable from a neuroprosthesis perspective because of the need to wear additional objects on the hand, but it may be a viable alternative if body-based imaging provides insufficient information.
- **The system must be able to accurately measure hand orientation.** Forearm pronation-supination is the primary movement used for orienting the hand, a critical component of any function requiring hand grasp (e.g., all of the acquisition tasks described earlier). Changes in hand orientation are made in a human user through rotation of the forearm about its long axis (i.e., pronation and supination). In neuroprosthesis applications, the speed with which this movement is performed does not need to be rapid – a forearm rotation speed comparable to the speed of the rest of the arm movement would be adequate. Although forearm rotation is a critical aspect of any grasping function, it has been difficult to measure in the past for several reasons:
    - Because pronation-supination rotations occur along the long axis of the forearm, global Cartesian movement at the forearm skin surface for a given internal angular rotation is small because of the short distance. This is in contrast to other joints like the elbow whose long body segment lengths (humerus proximally and forearm distally) produce large Cartesian motions for a given joint rotation. Measurement by markers placed on the skin surface is therefore not particularly sensitive to the internal bone rotations and prone to inaccuracies.
    - To overcome the sensitivity problems described above, devices for measuring forearm orientation can use long cantilevers that mechanically magnify the Cartesian movement

for a given forearm rotation. This is impractical for a neuroprothesis, however, since such devices would in appropriately interfere with activities of daily living. This is especially true for devices attached to the hand.

  - Measuring bone rotations in the forearm (i.e., the radius relative to the ulna) using markers attached to the skin is prone to errors because of large relative movements between the skin and the bones. Thus, any measurement device attached to the forearm is susceptible to slips that can introduce significant errors into the measurements. The REFES system has the potential to overcome this problem by directly imaging bony landmarks that are relatively prominent (e.g., the shape of the hand). However, the imaging procedures will need to be able to operate for different hand configurations (i.e., different degrees of opening and closing).

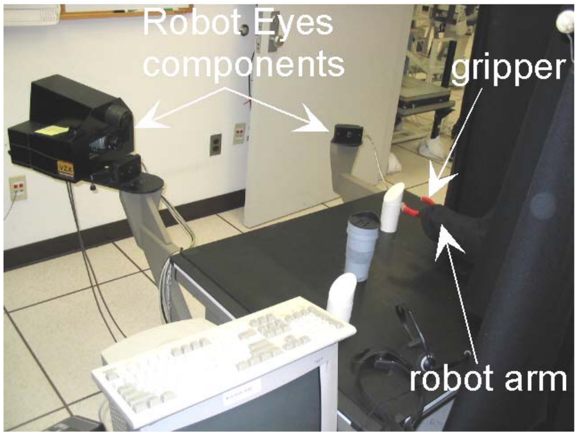
- **A robust method for dealing with situations where the endpoint is obscured must be developed.** The hand can be obscured in several ways during normal operation of the REFES system, e.g., by fixed objects during movement along a trajectory, by the user's own arm, or by objects moving into the scene. Discontinuous or erroneous endpoint position information due to camera blockage could lead to the robot simulator moving in a dangerous manner. Furthermore, movement of the robot simulator and/or a human with a real neuroprosthesis could result in collision with objects in the workspace, with the potential for spills. Solutions to this problem could include redundant imaging (e.g., hand and arm; skeletal landmarks and color variations; multiple artificial markers) so that endpoint position can be estimated even if the hand is partially obscured, and/or a control algorithm that detects blockage and interrupts the movement until a valid signal is reacquired. This additional intelligence (e.g., simply freezing the movement until a valid signal is obtained or extrapolating the currently obscured position based on previous movement state) could be included in either the REFES system or in the neuroprosthesis.

In summary, the basic feasibility for REFES to provide position signals for use in neuroprosthesis control has been demonstrated. There are a number of significant advantages to such a scheme if it can be practically realized (see bullet list on page 21). At the conclusion of this project, this approach was not sufficiently developed to safely implement on the Staubli robot setup, although it appears that the remaining issues are tractable. If resolved, the REFES system has the potential to provide a practical, contactless method for measuring endpoint position AND hand orientation. Such a capability would be a significant contributor to the field of neuroprostheses.

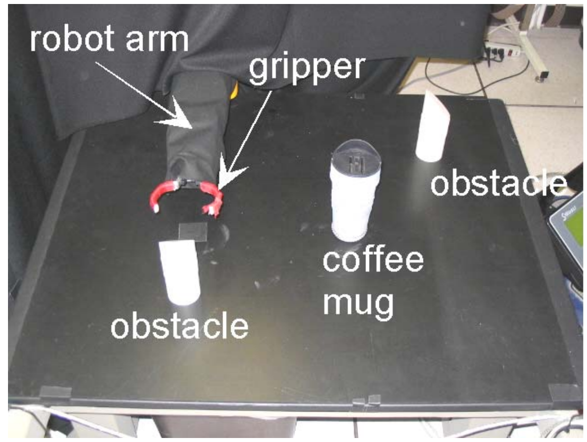
Task b: evaluate performance for range of motion

Demonstration: REFES -FES system

The REFES / neuroprosthesis combined system was demonstrated by having an able-bodied subject control the Staubli robot in a series of tasks that emulated typical functional tasks that would be performed by an individual with high tetraplegia in their activities of daily living. The following paragraphs will describe the particular methods used and then summarize the results. Several video files that were obtained during these trials are included on the CD that accompanies this report.



(a)



(b)

Figure 9: Photographs of the apparatus used in the human demonstration trials. Part (a) illustrates the REFES system components mounted off the front of the table used to define the horizontal workspace used in these experiments. Part (b) shows that robot arm and gripper more clearly, and also shows the 3 objects used in these tests: a coffee mug and two while cylinders that were used as obstacles.

Methods

The configuration of the REFES system relative to the Staubli robot is illustrated in Figure 9. Figure 9 (a) shows the basic components of the REFES system and how they were mounted relative to the Staubli Robot. The imaging system and spotting cameras of the REFES system were mounted off the front of the table that defined the available workspace, facing in towards the Staubli robot arm. The distal segments of the robot arm and the gripper (in red) are visible in this picture but not clear. Figure 9 (b) more clearly illustrates the robot arm and gripper. Also illustrated are the objects that can be recognized by the REFES system. The coffee mug and two white cylinders were imaged at SIS and entered into the REFES database for use in these tests at CWRU. Most of the robot arm is draped in a black cloth to reduce reflections during the initial calibration procedures during which the scene is imaged. In Figure 10, the black drapes around the robot arm have been pulled back to illustrate how the Staubli arm is configured in these tests.

Figure 10: The black drapes that covered the Staubli robot in these trials have been pulled back to illustrate the location of the robot relative to the workspace.

As described above, a voice recognition system and a head tracker mouse emulator were used as the interface between the human user and the REFES system. The head tracker was used to move the cursor around on the screen to

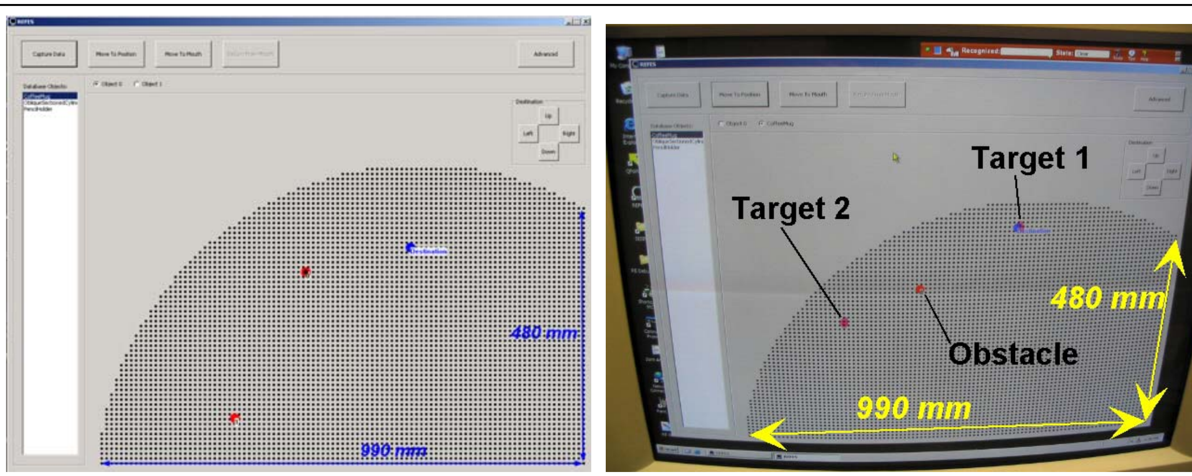
allow the user to specify different actions. The voice recognition system was used in these trials only to specify a mouse click. A photograph of a user wearing the head tracker system and a microphone for the voice recognition system is shown in Figure 11.

The actions that the user had to perform in order to make the appropriate commands to the REFES system can be understood by examining the REFES graphical interface illustrated in Figure 12. Part (a) is a screen dump of the REFES interface for some arbitrary situation.

Figure 11: A user wearing the visor-like head tracker mouse emulator and a standard computer microphone for the voice

In this case, the position of two objects (0 and 1) are indicated by red dots on the screen and the desired location of object 0 (the coffee mug) is indicated by the blue dot labeled "destination". The user has a number of options for commanding the REFES system:

1. Select different objects for manipulation. This is done by clicking on the "radio button" of one of the listed objects (in this case object 0 or object 1), which are located in the top left of the screen just underneath the button labeled "Move to Position. This is accomplished by using the head tracker to move the cursor over the desired button and then selecting it by speaking "click", which is recognized by the voice recognition system as meaning "left



(a) (b)

Figure 12: Graphical user interface provided by the REFES system. Part (a) is a screen dump of the basic interface used. Part (b) illustrates the location of the two targets used in the point-to-point movement trials in this demonstration. Target 1 would be located in front of the shoulder at arm's length while Target 2 would be located nearer the body and close to the midline. The straight-line distance between Target 1 and Target 2 was 482 mm. The obstacle was present for one series of point-to-point movements and was absent in the other set of point-to-point movements.

mouse click". Then by using the same mouse-control style, the name of the selected object has to be chosen too from the list of different object names that is located on the left-hand side of the screen.

2. Select an arbitrary location within the workspace to which the selected object is moved. This is accomplished by moving the by using the head tracker to move the cursor to the desired location on the screen and then selecting it by speaking "click", which is recognized by the voice recognition system as meaning "left mouse click".
3. Command an action by the REFES system. There are several options that are indicated by the square buttons listed across the top of the screen:
   - **Capture data.** Selecting this command activates the REFES system to image the workspace and identify objects within the workspace. This is done at the beginning of a session as described above.
   - **Move to position.** This commands the REFES system to move to a location ("destination") that was previously selected by the user in step #2. The REFES system will then automatically command the robot to move to the object selected for manipulation in step #1, grasp it with the gripper, and then move it to the destination selected. The trajectory computed by the REFES system will avoid any obstacles in the scene.
   - **Move to mouth.** This commands the REFES system to move to a pre-set location that would be close enough to a user's mouth to allow functions like drinking with a straw, brushing teeth, etc. The REFES system will automatically command the robot to move to the object selected for manipulation in step #1, grasp it with the gripper, and then move it to the "mouth" destination, avoiding any obstacles in the scene
   - **Return from mouth.** This is the reverse operation as "Move to mouth". When the user is finished with the object nears his or her mouth, this command causes the REFES system to command to the robot to return the object to its original location and release it from the gripper, again avoiding any obstacles.
4. Command incremental movements of the gripper. The left-right and up-down arrows located in the top right of the computer screen allow the user to move in 1 cm increments each time they are clicked upon. This would give the user the ability to fine-tune their final gripper position. These actions would be accomplished by moving the by using the head tracker to move the cursor over the arrow indicating the desired movement then selecting it by speaking "click", which is recognized by the voice recognition system as meaning "left mouse click".

During the tests performed in this demonstration, the following specific tasks were performed:

1. The user put on the head tracker mouse system and the microphone for the voice recognition system.
2. The REFES system, Master Controller, and Staubli robot were prepared for use as described above.
3. The voice recognition system (QPointer) was activated.
4. The user initiated the initial calibration procedures during which the REFES system scans the workspace and identifies objects contained within it. (See the video included in file "RE_startup_procedure" on the accompanying CD).
5. The user then commanded the Staubli robot to acquire a coffee mug that was in the workspace, bring it to the mouth for simulated drinking, and then replace it back in the workspace. This was repeated several times, during which the movement time was

measured using a stopwatch and recorded. (See the video included in the file "RE_time_to&from_mouth_final" on the accompanying video).

6. The user then commanded the Staubli robot to move the coffee mug back and forth from one specified location and to a second specified location. For these trials, no obstacle was present. Both of these locations were indicated to the user by small red dots drawn on the computer screen using a marker pen. These two locations are indicated in Figure 12(b) and represent movements by a real user from a distant position in front of the shoulder to a closer position nearer the midline of the body. The total movement size was 482 mm. These movements were repeated several times, during which the movement time was measured using a stopwatch and recorded. (See the video included in the file "RE_point-to-point_no_obstacles" on the accompanying video).
7. The user then commanded the Staubli robot to move the coffee mug back and forth from the same locations used in #6 except that an obstacle (the white cylinder) was placed between the target locations. The location of the obstacle is labeled "obstacle" in Figure 12. These movements were repeated several times, during which the movement time was measured using a stopwatch and recorded. (Unfortunately, no video record of these measurements exists because the end of the tape was reached just prior to these trials. This was not noticed by the experimenters. The trials proceeded very similarly to those performed in #6 above, except the movement times were longer because the robot took a longer trajectory to avoid the obstacle.)

Results

<u>Subjective impressions</u>:

- The user interface to the REFES system was reasonably intuitive and easy to use. In particular, the head tracker mouse system required no training and could very accurately move the mouse cursor across the entire computer screen. The voice recognition system performed adequately, although it did not always recognize the mouse click command on the first attempt. It should be noted that the voice recognition system was not specifically trained for the tested user because of time constraints. Such training would undoubtedly improve its performance.

Trial	Robot to Object	Move	Total	Time (sec) Return to Table
1	8.8	7.3	16.1	15
2	9.8	7.6	17.4	14
3	10.4	7.3	17.7	15
4	10.3	7.6	17.9	14
5	10.4	7.4	17.8	22
Average	9.9	7.4	17.4	16
StdDev	0.6	0.1	0.7	3.0
% of action	57.2%	42.8%		

Table 3: Movement time and its components measured during movements from the table top to the mouth and back. "Robot to object" is the time from when the user was instructed to make a command until the robot grasped the coffee mug. "Move" is the time required to then move from the table to the mouth, and "Total" is the sum of "Robot to Object" and "Move". "Return to Table" is the total time needed to return from the mouth location to the original location on the table. All times are given in seconds.

- The movement times for a particular movement were very consistent from trial to trial.
- The recorded movement times were functionally reasonable. That is, they were fast enough to be very useful in a functional situation.
- A significant fraction of the total movement time occurred after the user had successfully indicated the desired action. Much of this was due to safety-related movement speed limitations of the robot.
- Obstacles in the workspace significantly increased movement time. This was seen as a small price to pay for relieving the user of the burden of negotiating through the obstacles on a moment-to-moment basis.

Quantitative results

As noted above, three different tasks were performed in these demonstrations. The first two of these tasks (moving a coffee mug to mouth and back, moving the coffee mug from point to point without obstacles) were captured on video. The files for these videos are included in the CD that accompanies this report.

Tables 3, 4, and 5 list the results from the three sets of trials performed for this demonstration. During the performance of each of the trials, one of the experimenters used a stopwatch to time the duration from a "go" signal until the robot had acquired the coffee mug, moved it to the desired location, and released it.

Table 3 lists the results obtained from the "table to mouth" and "mouth to table" trials. The total time required for the user to command the table-to-mouth movement and for the robot to actually make the commanded movement (the column labeled "Total" in Table 3) was very consistent, with a mean time across 5 trials of 17.4 seconds with a standard deviation of just 0.7 seconds. Approximately 57% of this total time was required for the user to command the movement and the robot to acquire the coffee mug (the column labeled "Robot to object" in Table 3). The remaining 43% of the time was required to move the mug up to the mouth (the column labeled "Move" in Table 3). It should be noted that much of the "total" task time was devoted to the actual motion of the robot. The average "Move" phase (from object to mouth) averaged 7.4 seconds and this time was almost entirely due to the relatively slow motion of the robot. The "Robot to object" phase (average duration of 9.9 seconds) included a similar movement phase from the "home position" to the object, so it is likely that only 2-3 seconds (i.e., 9.9-7.4) of the average "total" time of 17.4 seconds was expended by the user specifying the object. Since we purposely limited the robot's speed in these trials to 5% of its maximum capability, it would be possible to significantly reduce the total task time by increasing the speed of the robot's motion. This is not reasonable to do, however, both because of safety considerations and because the slower speeds used are much more comparable to the speed we expect from a paralyzed human arm under the control of a neuroprosthesis. Furthermore, the average task time (17.4 seconds) is very reasonable for the functional task being simulated. The "mouth to table" movement lacked the initial acquisition phase because the robot already held the coffee mug in the "mouth" position. Thus, these movements (labeled "Return to Table in Table 3) had only one component and were typically 2-3 seconds shorter than the "table to mouth" movement. Note that the user had difficulty with the voice recognition system in trial 5, producing a 22 second return to table movement that was much longer than any of the other 4 trials.

Table 4 illustrates the movement times recorded during a series of point-to-point movements from Target 2 to Target 1 and from Target 1 back to Target 2, in both cases with no obstacle. This was a more demanding task than "Robot to Mouth", so the movement was divided into three components. "Selection" indicates how long it took the user to specify the desired location and then command the REFES system to start the motion. "Robot to object" is the time needed for REFES to specify the trajectory and for the robot to move to and acquire the coffee mug. "Move" is the time from when the coffee mug was acquired until it was moved to the new location and released. "Total" is the sum of

	From Target 2 to Target 1				From Target 1 to Target 2			
Trial	Selection	Robot to Object	Move	Total	Selection	Robot to Object	Move	Total
1	7.2	17.5	6.8	31.5	7.8	5.3	8.5	21.6
2	6.5	8.3	6.0	20.8	5.3	4.7	7.3	17.3
3	4.3	8.4	6.8	19.5	6.3	5.9	6.8	19.0
4	6.4	8.4	5.8	20.6	5.5	5.5	6.5	17.5
5	6.0	8.9	5.8	20.7	4.1	5.9	6.5	16.5
6	5.5	8.6	5.7	19.8	4.2	5.9	6.6	16.7
7	5.6	8.1	6.1	19.8	4.5	5.5	6.3	16.3
8	5.8	8.3	6.4	20.5	4.7	5.4	6.3	16.4
Average	5.9	9.6	6.2	21.7	5.3	5.5	6.9	17.7
StdDev	0.8	3.0	0.4	3.8	1.2	0.4	0.7	1.7
% of action	27.3%	44.2%	28.5%		30.0%	31.2%	38.8%	

Table 4: Movement time and its components measured during point-to-point movements from Target 1 to Target 2 and from Target 2 back to Target 1, in both cases with no obstacle. This was a more demanding task than "Robot to Mouth", so the movement was divided into three components. "Selection" indicates how long it took the user to specify the desired location and then command the REFES system to start the motion. "Robot to object" is the time needed for REFES to specify the trajectory and for the robot to move to and acquire the coffee mug. "Move" is the time from when the coffee mug was acquired until it was moved to the new location and released. "Total" is the sum of "Selection", "Robot to Object" and "Move". All times are given in seconds. Note that the "Robot to Object" time for the Target 1 to Target 2 movement was significantly longer for trial 1 than any other trial.

"Selection", "Robot to Object" and "Move". These movement times (both the total time and each of the movement components) were again quite consistent from trial to trial, with one exception. The movement from Target 2 to Target 1 in Trial 1 was considerably longer than for any other trial. Extra computations performed by REFES for the first movement after imaging the scene artificially elevated this movement time. If this trial is omitted, the average movement time decreases and the variability decreases substantially (from 21.7 ± 3.8 seconds to 20.2 ± 0.5 seconds).

Movement from Target 1 to Target 2 was about 10% faster than movements from Target 2 to Target 1. This was due to the fact that the robot started each movement from the "home" position, which was located much closer to Target 1 than Target 2. Excluding trial 1 for the Target 2 to Target 1, the "Robot to Object" movement component was 8.4 ± 0.2 seconds for Target 2 to Target 1, while it was only 5.5 ± 0.4 seconds for Target 1 to Target 2 movements.

For both movement directions, about 30% of the total movement time was due to the user interface with the REFES system, i.e., was the time required by the user to move the mouse cursor to the target, click on this destination, and then move to "Move to Position" and click. The rest of the time was internal to the REFES system and the robot, i.e., the time required to move to the object, acquire it, and move it to the new location.

Trial	Time (sec) T2 to T1	T1 to T2
1	33	23
2	30	21
3	29	23
4	30	20
5	28	22
6	28	20
Average	29.7	21.5
StdDev	1.7	1.3

Table 5: Total movement time measured during point-to-point movements from Target 1 to Target 2 and from Target 2 back to Target 1, in this case with an obstacle located between the targets as illustrated on Figure 10.

Table 5 illustrates the total movement time for the same movements made in Table 4 except that an obstacle was included. As noted previously, only the total movement times were recorded because the video was inadvertently not obtained. The expected results would be that the user interface time should be similar to those obtained for the non-obstacle trials. However, the need for the REFES system to avoid the obstacle will increase any movements towards or away from Target 2. Thus, it is likely that the "Robot to Object" time for movements from the "home" position to Target 1 would be unchanged by the obstacle, but that all other times would be increased by the longer and more complicated trajectory required to avoid the obstacle when moving to or away from Target 2. This is reflected in the results. The Target 1 to Target 2 trajectory times required only on obstacle avoidance trajectory (when moving from Target 1 to Target 2). These times were greater than without the obstacle (21.5 ± 1.3 versus 17.7 ± 1.7). However, they were substantially less (21.5 ± 1.3 versus 29.7 ± 1.7). than the Target 2 to Target 1 movement that required two obstacle avoidance trajectories (when moving to first acquire the mug at Target 2 and then when moving from Target 2 to Target 1).

Milestone Five: Final report

This report satisfies Milestone Five.

Summary and conclusions

Investigators and staff within the Cleveland FES Center of Case Western Reserve University have completed the objectives of their contract with Spatial Integrated Systems to demonstrate the potential for the REFES system to serve as an effective user interface for a neuroprosthesis for individuals with high tetraplegia resulting from high cervical spinal cord injury. The results obtained under this contract include:

- The specification of the types of movements that can be realistically restored and functionally important for individuals with high tetraplegia.
- The selection of preferred user interface methods between neuroprosthesis users with high tetraplegia and the REFES system.
- A robotic apparatus that can be used to simulate a paralyzed arm for the purposes of developing neuroprosthesis command interfaces was designed, fabricated, and demonstrated.
- The REFES system was successfully interfaced with the Master Controller of the neuroprosthesis system.
- The REFES system successfully created and understood 3D environment of the robot working space and was able to identify and locate 3D objects.

- The REFES system successfully controlled the motion of the robot simulator and avoided possible static or moving obstacles, successfully demonstrating the interface between REFES and the neuroprosthesis. This will greatly facilitate the introduction of the REFES system into real neuroprosthesis testing when implanted human subjects are available in 12-18 months. Note that the robot used at CWRU is significantly different from the one in use at SIS, demonstrating the flexibility of the REFES system
- In conjunction with Ardiem and SIS, the basic feasibility of providing measurements of endpoint position of the human arm was demonstrated.
- Functional use of the REFES /neuroprosthesis system was demonstrated by a human user commanding the robot to perform three different tasks. The system was found to be straightforward to use and the movement times for the various tasks were well within functionally tolerable ranges.

We conclude that the REFES system is likely to play a significant role in the continued development of a neuroprosthesis for high tetraplegia. As noted several times in the above report, the REFES interface has several major positive attributes that are not seen in alternative command interfaces. In particular, the REFES -based approach removes a significant fraction of the tedium and effort from the user because it provides all of the object recognition and low level trajectory planning, obstacle identification and avoidance, robot working space environment monitoring, allowing the user to focus on the high-level goal (e.g., “pick up the cup”) rather than all of the details. The REFES system “knows” the objects in the scene and defines trajectories that automatically approach the object in an appropriate manner (e.g., towards the handle of the mug). We fully expect to implement and test the REFES interface with human subjects when real neuroprostheses are implemented in human subject in Spring 2005.

These same attributes that make the REFES interface very attractive for neuroprosthesis applications to high tetraplegia also suggest several other related applications. In particular, the command interfaces for rehabilitation robots that are commonly used by individuals with high tetraplegia, muscular dystrophy, amyotrophic lateral sclerosis (i.e., “Lou Gehrig's disease”), and other neurological and musculoskeletal disease are currently surprisingly ineffective. Given the existing ability of REFES to control different types of robots, interfacing this system to a rehabilitation robot should be relatively straightforward. The additional hardware beyond the robot itself that is added to the wheelchair should be acceptable with some minor modifications to the REFES imaging system. We believe that the REFES command interface would be far superior to existing systems and could significantly increase the market for rehabilitation robots.

Appendix I: Staubli robot kinematic equations

Six variables, defined by constants in the block diagram, constitute the inputs for the program. Three of the inputs - X, Y, and Z - correspond to the location of the "object" in space. The global, Cartesian frame of reference, along whose axes the x, y, and z coordinates lie, has its origin at the base of the arm. X points directly out, Y points to the right, and Z points straight down. These three variables define a vector from the global origin to the object, called the "position vector". The other three inputs - yaw, pitch and roll - describe the Euler angles that define the necessary orientation of the manipulator. Yaw is the rotation about the OZ axis, pitch is the rotation about the OV axis (where V is the rotated Y axis), and roll is the rotation about the OW axis (where W is the rotated Z axis).

The program first calculates the "rotation matrix", a matrix composed of three vectors of three components each. It is defined as follows:

$$\begin{bmatrix} \cos\phi\cos\theta\cos\psi - \sin\phi\sin\psi & -\cos\phi\cos\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\sin\theta \\ \sin\phi\cos\theta\cos\psi + \cos\phi\sin\psi & -\sin\phi\cos\theta\sin\psi - \cos\phi\cos\psi & \sin\phi\sin\theta \\ -\sin\theta\cos\psi & \sin\theta\sin\psi & \cos\theta \end{bmatrix} = \begin{bmatrix} \vec{n} & \vec{s} & \vec{a} \end{bmatrix}$$

(Equation 1)

In Equation 1, phi (ϕ) is yaw, theta (θ) is pitch, and psi (ψ) is roll (Fu *et al* 23-24). **n** is the "normal vector", which is perpendicular to the fingers of the robot arm. **s** is the "sliding vector", which points in the direction of the motion of the fingers. **a** is the "approach vector", which points from the wrist to the object and is perpendicular to the "palm" of the manipulator. **n**, **s**, and **a** serve as a means for relating the frame of reference of the manipulator to the global frame of reference (Fu *et al* 43).

Next, the program uses **a** and the global coordinates of the object to calculate the necessary coordinates of the Wrist. It does so using the equation

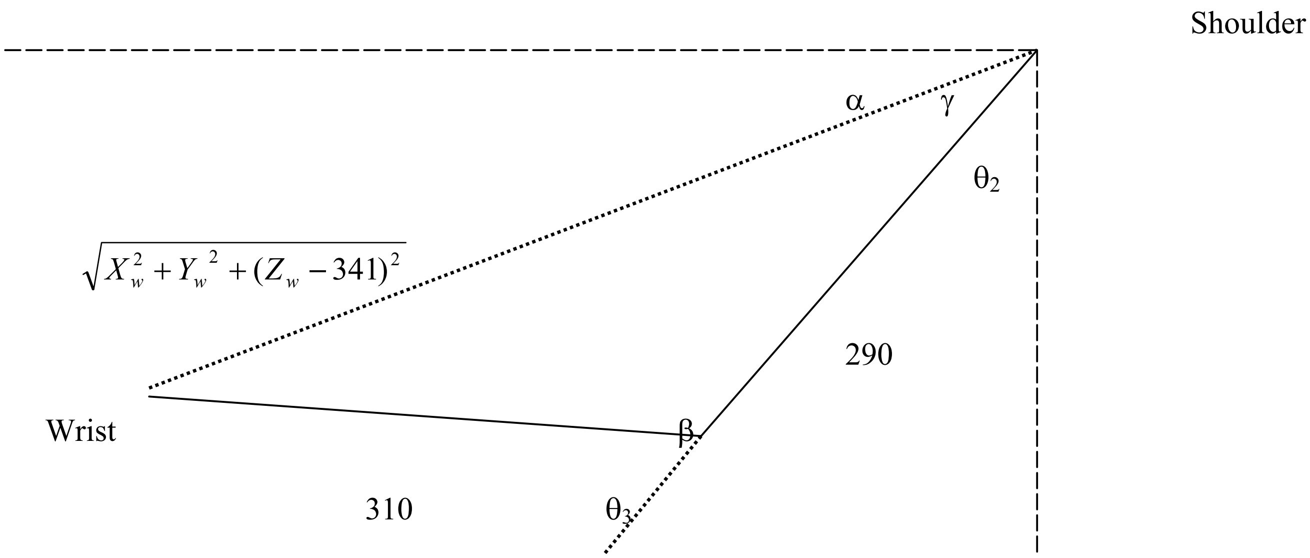
$$\vec{p}_{wrist} = \vec{p}_{object} - d_6\vec{a} \qquad \text{(Equation 2)}$$

This is basically just vector addition, where $\mathbf{p}_{\text{object}}$ is the position vector of the object as defined by the inputs, d_6 is the length of the manipulator from the Wrist to the middle of the gripper, and $\mathbf{p}_{\text{wrist}}$ is the position vector of the Wrist (Fu *et al* 63).

Knowing the X and Y coordinates of the Wrist, the program determines the angle θ_1 through which the Waist must rotate. Since the robot arm is incapable of lateral motion, the Waist alone determines the angular distance from the X axis of the Wrist. Therefore, θ_1 can be determined by

$$\theta_1 = \arctan\left(\frac{x_{wrist}}{y_{wrist}}\right) \qquad \text{(Equation 3)}$$

θ_2 and θ_3 are more complex to find, since they are used together to determine both the distance from the global origin and the Z coordinate of the Wrist. However, finding them is simply a matter of trigonometry. Figure 1 shows the X'Z' plane, where the ' indicates that the frame of reference has been rotated by θ_1.



(Figure 1)

The line from the Shoulder to the Wrist has been drawn to create a triangle, thus enabling the use of the sine and cosine laws. First, we can use simple trigonometry to find α, where $\alpha = \arctan (Z_w-341)/Y_w$ (note: we use Z_w-341 rather than Z_w because the shoulder is 341 mm away from the global origin on the Z axis.) Then, using the cosine law, we can define β as $\arccos\left(\frac{-Y_w^2 - X_w^2 - Z_w^2 + 290^2 + 310^2}{179800}\right)$. This value can be used in the sine law to find γ = $\arcsin\left(\frac{310\sin\beta}{\sqrt{Y_w^2 + X_w^2 + (Z_w - 341)^2}}\right)$. Combining these and the concept of complementary and supplementary angles, we can define θ_2 and θ_3 as

$$\theta_2 = 90 - \alpha - \gamma \qquad \text{(Equation 4)}$$

$$\theta_3 = 180 - \beta \qquad \text{(Equation 5)}$$

Using the solutions provided by Fu *et al*, we can write equations that solve for the final three joint angles. These joints ensure that the orientation of the manipulator will correspond to the roll, pitch, and yaw specified by the user. The equations for the joint angles for the Forearm and Wrist joints are simply:

$$\theta_4 = \arctan\left(\frac{C_1 a_y - S_1 a_x}{C_1 C_{23} a_x + S_1 C_{23} a_y - S_{23} a_z}\right) \qquad \text{(Equation 6)}$$

$$\theta_5 = \arctan\left(\frac{(C_1 C_{23} C_4 - S_1 S_4) a_x + (C_1 C_{23} C\theta_4 - S_1 S_4) a_y - C_4 S_{23} a_z}{C_1 S_{23} a_z + S_1 S_{23} a_y + C_{23} a_z}\right) \qquad \text{(Equation 7)}$$

$$\theta_6 = \arctan\left(\frac{(-S_1 C_4 - C_1 C_{23} S_4) n_x + (C_1 C_4 - S_1 C_{23} S_4) n_y + S_4 S_{23} n_z}{(-S_1 C_4 - C_1 C_{23} S_4) s_x + (C_1 C_4 - S_1 C_{23} S_4) s_y + S_4 S_{23} s_z}\right) \qquad \text{(Equation 8)}$$

where $C_i=\cos\theta_i$, $S_i=\sin\theta_i$, $C_{ij}=(\cos\theta i+\theta_j)$, and $S_{ij}=\sin(\theta_i+\theta_j)$ (Fu *et al* 71-72).

The program checks to make sure of a number of things. First of all, two values of θ_2 are produced, and the one that is closer to zero is selected, since this is the one that is feasible with human anatomy. Also, various constraints are enforced to ensure that the robot arm behaves like a human arm. For example, the magnitude of the position vector $\mathbf{p}_{object}$ must be less than 800, and the roll, pitch, and yaw of the orientation vector must be within a certain range of values. If the program finds that any of these constraints is not met, it outputs a value of $5x10^7$ for every joint angle to show the user that the inputs were unusable.

It is important to note the difference between the intuitive frame of reference and that which was used for the program. Under normal circumstances, a person will consider their coordinate axes to be aligned in such a way that the X axis will point to the side right side, the Y axis will point forward, and the Z axis will point up. Fu *et al* modified this assessment so that the X axis points forward and the Y axis points to the left (Fu *et al* 37). Their equations are tailored to the normal use of a PUMA-style robot arm, which involves the robot being mounted to the floor. We, however, have mounted the arm upside down, and so we had to invert the coordinate axes, as described in the first paragraph.

Also, while Fu *et al* set up their rotation matrix in such a way that the manipulator was pointing straight up when pitch, roll, and yaw were 0, it was convenient for our purposes to set this base position to pointing out parallel to the x axis. Thus, to convert our rotations to fit theirs, we add 90° to the pitch angle (notice how we add, not subtract, because of the reasons that were explained in the previous paragraph).

References

Angeles, Jorge. Fundamentals of Robot Mechanical Systems: Theory, Methods, and Algorithms. New York: Springer, 2003.

Fu, K. S. *et al*. Robotics: Control, Sensing, Vision, and Intelligence. New York: McGraw Hill Book Company, 1897

Staubli robot kinematics code

```
procedure main()
num i
begin
  do
    cls()
    //resetMotion()
    //
    //Read in serial port
    for i=0 to 21
      raw_in[i]=portSerial
    endFor
    //call display_input()
    //
    //Find sync byte
    for i=0 to 21
      if raw_in[i]==250
        syncbyte=i
```

```
  endIf
endFor
//
//Restore byte arrangement
call alignbytes()
//call display_splits()
//
//Reconstruct joint angles
call build_angles()
//
//Display output
//call display_final()
//
//Check for gripper toggle
if split_angles[19]!=gripper_flag and error==0
  if split_angles[19]==1
    close(flange)
    delay(1)
    gripper_flag=1
  else
    call act()
    open(flange)
    gripper_flag=0
  endIf
endIf
//
//Check for pause/resume/stop
if split_angles[20]==220
  stopMove()
endIf
if split_angles[20]==222
  restartMove()
endIf
if split_angles[20]==225
  stopMove()
  resetMotion()
endIf
//
//Move arm to location
call act()
//
//
//delay(0.075)
//resetMotion()
//
loops=loops+1
if loops>10
  //resetMotion()
  loops=0
endIf
//
//find joint angles and transmit back
location=herej()
call configure_outpu()
call send_out()
call display_output()
//
```

```
  until (bIn0==true)
end


procedure act()
joint now
begin
  now.j1=jtangles[0]
  now.j2=jtangles[1]
  now.j3=jtangles[2]
  now.j4=jtangles[3]
  now.j5=jtangles[4]
  now.j6=jtangles[5]
  movej(now,flange,nom_speed)
end

procedure alignbytes()
num i
num j
begin
  switch syncbyte
    case 0
      for i=0 to 21
        split_angles[i]=raw_in[i]
      endFor
    break
    case 1
      for i=1 to 21
        split_angles[i-1]=raw_in[i]
      endFor
      split_angles[21]=raw_in[0]
    break
    case 2
      for i=2 to 21
        split_angles[i-2]=raw_in[i]
      endFor
      for j=20 to 21
        split_angles[j]=raw_in[j-20]
      endFor
    break
    case 3
      for i=3 to 21
        split_angles[i-3]=raw_in[i]
      endFor
      for j=19 to 21
        split_angles[j]=raw_in[j-19]
      endFor
    break
    case 4
      for i=4 to 21
        split_angles[i-4]=raw_in[i]
      endFor
      for j=18 to 21
        split_angles[j]=raw_in[j-18]
      endFor
    break
    case 5
```

```
    for i=5 to 21
      split_angles[i-5]=raw_in[i]
    endFor
    for j=17 to 21
      split_angles[j]=raw_in[j-17]
    endFor
  break
  case 6
    for i=6 to 21
      split_angles[i-6]=raw_in[i]
    endFor
    for j=16 to 21
      split_angles[j]=raw_in[j-16]
    endFor
  break
  case 7
    for i=7 to 21
      split_angles[i-7]=raw_in[i]
    endFor
    for j=15 to 21
      split_angles[j]=raw_in[j-15]
    endFor
  break
  case 8
    for i=8 to 21
      split_angles[i-8]=raw_in[i]
    endFor
    for j=14 to 21
      split_angles[j]=raw_in[j-14]
    endFor
  break
  case 9
    for i=9 to 21
      split_angles[i-9]=raw_in[i]
    endFor
    for j=13 to 21
      split_angles[j]=raw_in[j-13]
    endFor
  break
  case 10
    for i=10 to 21
      split_angles[i-10]=raw_in[i]
    endFor
    for j=12 to 21
      split_angles[j]=raw_in[j-12]
    endFor
  break
  case 11
    for i=11 to 21
      split_angles[i-11]=raw_in[i]
    endFor
    for j=11 to 21
      split_angles[j]=raw_in[j-11]
    endFor
  break
  case 12
    for i=12 to 21
```

```
    split_angles[i-12]=raw_in[i]
  endFor
  for j=10 to 21
    split_angles[j]=raw_in[j-10]
  endFor
break
case 13
  for i=13 to 21
    split_angles[i-13]=raw_in[i]
  endFor
  for j=9 to 21
    split_angles[j]=raw_in[j-9]
  endFor
break
case 14
  for i=14 to 21
    split_angles[i-14]=raw_in[i]
  endFor
  for j=8 to 21
    split_angles[j]=raw_in[j-8]
  endFor
break
case 15
  for i=15 to 21
    split_angles[i-15]=raw_in[i]
  endFor
  for j=7 to 21
    split_angles[j]=raw_in[j-7]
  endFor
break
case 16
  for i=16 to 21
    split_angles[i-16]=raw_in[i]
  endFor
  for j=6 to 21
    split_angles[j]=raw_in[j-6]
  endFor
break
case 17
  for i=17 to 21
    split_angles[i-17]=raw_in[i]
  endFor
  for j=5 to 21
    split_angles[j]=raw_in[j-5]
  endFor
break
case 18
  for i=18 to 21
    split_angles[i-18]=raw_in[i]
  endFor
  for j=4 to 21
    split_angles[j]=raw_in[j-4]
  endFor
break
case 19
  for i=19 to 21
    split_angles[i-19]=raw_in[i]
```

```
      endFor
      for j=3 to 21
        split_angles[j]=raw_in[j-3]
      endFor
    break
    case 20
      split_angles[0]=raw_in[20]
      split_angles[1]=raw_in[21]
      for i=2 to 21
        split_angles[i]=raw_in[i-2]
      endFor
    break
    case 21
      split_angles[0]=raw_in[21]
      for i=1 to 21
        split_angles[i]=raw_in[i-1]
      endFor
    break
  endSwitch
end
```

procedure build_angles()

```
num hundreds
num i
num j
begin
  error=0
  //
  //check for misplaced sync byte
  for i=1 to 21
    if split_angles[i]>245
      error=1
    endIf
  endFor
  //
  //Check for sync byte at top of array
  if split_angles[0]!=250
    error=1
  endIf
  //
  //Checksum
  checksum=0
  for j=1 to 20
    checksum=checksum+split_angles[j]
  endFor
  checksum=roundDown(checksum/6)
  if checksum!=split_angles[21]
    error=1
  endIf
  //
  //Check for found errors
  if error==0
    for j=0 to 5
      hundreds=split_angles[(j*3)+1]*100
      jtangles[j]=split_angles[(j*3)+2]+(split_angles[(j*3)+3]/100)
      jtangles[j]=hundreds+jtangles[j]-180
```

```
    endFor
    jtangles[2]=jtangles[2]+90
  else
    for i=0 to 5
      jtangles[i]=jtmemory[i]+jtmemory[i]-jtmemory2[i]
    endFor
  endIf
  //
  //
  //Set joint angle memories
  for j=0 to 5
    jtmemory[j]=jtangles[j]
    jtmemory2[j]=jtmemory[j]
  endFor
end
```

procedure configure_outpu()

```
num hundreds
num i
num temp_loc[6]
begin
  //Record current joint angles
  //and offset for tansmition
  temp_loc[0]=location.j1+180
  temp_loc[1]=location.j2+180
  temp_loc[2]=location.j3+90
  temp_loc[3]=location.j4+180
  temp_loc[4]=location.j5+180
  temp_loc[5]=location.j6+180
  //
  //Split angles
  for i=0 to 5
    angle_out[3*i]=roundDown(temp_loc[i]/100)
    angle_out[(3*i)+1]=roundDown(temp_loc[i])-(angle_out[3*i]*100)
    angle_out[(3*i)+2]=round(100*(temp_loc[i]-roundDown(temp_loc[i])))
  endFor
  //
  //Record current endpoint location
  //and offset for transmision
  hereandnow=here(gripper,world)
  endpoint=hereandnow.trsf
  temp_loc[0]=(endpoint.x+665)/2
  temp_loc[1]=(endpoint.y+665)/2
  temp_loc[2]=(endpoint.z+665)/2
  temp_loc[3]=endpoint.rx+180
  temp_loc[4]=endpoint.ry+180
  temp_loc[5]=endpoint.rz+180
  //
  for i=0 to 5
    points_out[3*i]=roundDown(temp_loc[i]/100)
    points_out[(3*i)+1]=roundDown(temp_loc[i])-(points_out[3*i]*100)
    points_out[(3*i)+2]=round(100*(temp_loc[i]-roundDown(temp_loc[i])))
  endFor
  //
end
```

procedure display_final()

```
begin
  //Display output
  put("Error State: ")
  putln(error)
  put("Syncbyte: ")
  put(syncbyte)
  put("        ")
  put("Sync: ")
  putln(raw_in[syncbyte])
  put("Rbt Chksm: ")
  put(checksum)
  put("     ")
  put("XPC Chksm: ")
  putln(split_angles[14])
  put("Jt1: ")
  putln(jtangles[0])
  put("Jt2: ")
  putln(jtangles[1])
  put("Jt3: ")
  putln(jtangles[2])
  put("Jt4: ")
  putln(jtangles[3])
  put("Jt5: ")
  putln(jtangles[4])
  put("Jt6: ")
  putln(jtangles[5])
  put("Gripper: ")
  putln(split_angles[13])
end

procedure display_input()
num i
begin
  for i=0 to 13
    put(raw_in[i])
    put("  ")
  endFor
  putln("")
  putln("")
end

procedure display_output()
num i
begin
  for i=0 to 10 step 2
    put(angle_out[i])
    put(".")
    putln(angle_out[i+1])
  endFor
end

procedure display_splits()
num i
begin
  for i=0 to 13
    put(split_angles[i])
    put("  ")
```

```
  endFor
  putln("")
  putln("")
end

procedure send_out()
num i
begin
  portSerial=250
  for i=0 to 17
    portSerial=angle_out[i]
  endFor
  for i=0 to 17
    portSerial=points_out[i]
  endFor
end

procedure start()
num i
begin
  open(flange)
  gripper_flag=0
  //
  //set starting position
  for i=0 to 5
    jtangles[i]=0
  endFor
  jtangles[2]=90
  //
  //initialize memories
  for i=0 to 5
    jtmemory[i]=jtangles[i]
    jtmemory2[i]=jtangles[i]
  endFor
  //
  //set blend parameters
  nom_speed.leave=10
  nom_speed.reach=10
  //
  //initialize joint reporting
  for i=0 to 11
    angle_out[i]=90
  endFor
  //
  //call main program
  call main()
end

procedure stop()
begin
  popUpMsg("Pending movement commands have been canceled")
  resetMotion()
end
```

Appendix II. Previously submitted progress reports

Summary of all products considered for user-to-REFES interface:

Voice Recognition Products

- QPointer Hands Free

- Completely hands-free computer control by voice. Intuitive operation of any application by voice and full voice control over the Windows environment. Includes all capabilities of QPointer Keyboard.
- web site: http://www.commodio.com/products_voice.html
- Manufacturer: Commodio
- Price: $189

- Dragon NaturallySpeaking7 Standard

- Dragon NaturallySpeaking® Standard let's you talk to your computer and your words instantly appear in letters, e-mails, instant messages and chat rooms. You can even surf the web by speaking! Dictate and edit in Microsoft® Word, Microsoft® Outlook® Express, America Online®, and Corel® WordPerfect® and virtually any Windows®-based program. It's fast, accurate, and easy to use.
- web site: http://www.scansoft.com/naturallyspeaking/home/
- Manufacturer: ScanSoft
- Price: $99

- IBM Via Voice

- Designed as a powerful productivity tool, ViaVoice for Windows Advanced Edition Release 10 provides enhanced ease-of-use features for dictation and voice command of PC and Internet applications. A new speech engine can provide exceptional accuracy. Advanced Edition now supports selected digital handheld recorders, and comes with a stereo headset microphone with inline volume and mute controls.
- web site: http://www-3.ibm.com/software/speech/windows/version10/advanced/index.shtml
- Manufacturer: IBM
- Price: $67

Head Pointers / Mouse Emulators

- HeadMaster Plus

- Prentke Romich's HeadMaster Plus™ is a head pointing system that takes the place of a mouse. Just move your head and the cursor moves on the screen. Puff on the tube to make selections. Mouse clicks can also be made by activating an external switch (sold separately) or by dwelling with a dwell software program (sold separately). It is the only head pointing system that tracks both lateral and rotational movement. Also available for HeadMaster is an optional Remote Adapter providing for wireless infrared use and an optional Laptop Adapter.
- web site: http://store.prentrom.com/catalog/prentrom/HM-3P
- Manufacturer: Prentke Romich
- Uses ultrasound
- Price: $995

HeadMouse

- Origin Instruments' HeadMouse™ is a head operated mouse. It has a sensor that tracks a tiny, reflective dot placed on your forehead or glasses. When you move your head, the cursor follows on the screen. Mouse clicks can be made by activating an external switch (sold separately) or by dwelling with a dwell software program (sold separately). When combined with an on screen keyboard, HeadMouse can completely replace a traditional mouse and keyboard. HeadMouse comes with a power adapter, a cable for serial connection and 50 disposable reflective dots. If you require a PS2, ADB or USB connection you must order a "Smart Cable" separately.
- web site: http://www.orin.com/access/ , or
  http://store.prentrom.com/catalog/prentrom/HE-S, or
  http://www.infogrip.com/product_view.asp?RecordNumber=116&sbcolor=%23CC9966&option=pointing&subcategory=13&CatTxt=Head+Controlled&optiontxt=Pointing
- Manufacturer: Origin Instruments
- Uses infrared
- Price: $1795

- Tracker 2000

- Tracker 2000 allows you to smoothly move the cursor on the computer simply by moving your head, regardless of your disability. Tracker 2000 sits on top of the computer and tracks a tiny reflective "dot" worn on your forehead or glasses. When you move your head, Tracker 2000 elegantly converts that into computer mouse movements.
- web site: http://store.prentrom.com/catalog/prentrom/TR2000
- Manufacturer: Madentec
- Uses infrared
- Price: $1595

- Tracker One

- Tracker One is a truly revolutionary head pointing device. Tracker One makes computer access even easier. It operates from the USB port on your computer or compatible AAC device and gives you both the freedom to be completely mobile without need of battery packs or power adapters. Tracker One incorporates all the dependability and functions you trust from Tracker2000 but has simplified your connection options.

- web site: http://store.prentrom.com/catalog/prentrom/TR1
- Manufacturer: Madentec
- Price: $895

  - Smart-Nav AT

- Natural Point's Smart-Nav AT mouse alternative gives you hands free control of a computer cursor. With a reflective dot on your forehead, it provides precise cursor control through simple head movements. Mouse clicks can be accomplished through a built in dwell clicking program or external switches (sold separately). You can also control the cursor with an included ring, allowing you to perform all typical mouse functions by simply aiming your finger at any point on your screen and clicking with your user defined Hot Keys. Included with Smart-Nav AT is an on-screen keyboard for hands free keyboarding. The Smart-Nav AT is a USB device and requires no external power. Just connect Smart-Nav AT to your computer and you're ready to go.
- web site: http://www.naturalpoint.com/prod/product.htm or http://www.infogrip.com/product_view.asp?RecordNumber=530&sbcolor=%23CC9966&option=pointing&subcategory=13&CatTxt=Head+Controlled&optiontxt=Pointing
- Manufacturer: Natural Point (product used to be called TrackIR)
- Uses infrared
- Price: $299

  - Tracer

- Boost Technology's Tracer is a mouse that you control with your head. Tracer gives mouse control to people with Quadriplegia, Cerebral Palsy, Muscular Dystrophy, Multiple Sclerosis, ALS, Carpal Tunnel Syndrome and any other disability where the user lacks the hand control to use a standard mouse but retains good head movement. Tracer uses a small gyroscope to sense the user's motion. The gyroscope communicates wirelessly with the computer. Because it's patented micro-gyroscope technology is remarkably precise - down to individual pixel resolution - anything that can be done with a mouse, you can do with Tracer. Draw. Surf. Design. Communicate. Connect.
- web site: http://www.infogrip.com/product_view.asp?RecordNumber=506&sbcolor=%23CC9966&option=pointing&subcategory=13&CatTxt=Head+Controlled&optiontxt=Pointing
- Manufacturer: Boost Technology
- Uses gyroscopes
- Price: $795

  - Miracle Mouse

- Miracle Mouse provides users with all the tools necessary to operate any Microsoft Windows application - hands free! Send and receive email from around the world, surf the Internet, make on-line purchases, use word processing to write documents, use spreadsheets to help balance your checkbook, create business presentations, design databases, play games (online or PC), listen to music through internet radio, find a career, or automate your home. Miracle Mouse comes with on-screen keyboards, word prediction/word completion, international word lists, user definable macro panels, visual enhancements, text-to-speech, automatic arrange & screen positioning.

- web site: http://www.maui-innovative.com/content/AssistiveTech.html
- Manufacturer: Maui Innovative Peripherals
- Price: $499

Eye Gaze Systems

- Eyegaze Communication System

- The Eyegaze System is a communication and control system for people with complex physical disabilities. You run the system with your eyes. By looking at control keys displayed on a screen, a person can synthesize speech, control his environment (lights, appliances, etc.), type, operate a telephone, run computer software, operate a computer mouse, and access the Internet and e-mail. Eyegaze Systems are being used to write books, attend school and enhance the quality of life of people with disabilities all over the world.
- web site: http://www.eyegaze.com/doc/ecs.htm
- Manufacturer: LC Technologies
- Price: $14,900

- Quick Glance 1

- Place the mouse pointer anywhere on the screen simply by looking at the desired location. Click with an eye blink, a hardware switch, or by staring (dwell). Combine Quick Glance with an on-screen keyboard to communicate with text or speech output. Various options for emulating mouse functions give accessibility to all Windows features including right clicking, dragging, and double clicking!
- web site: http://www.eyetechds.com/homepage.html
- Manufacturer: EyeTech Digital Systems
- Price: $ 3,950

July 2, 2003 Progress Report

Progress:

Further refinements to the point-to-joint transformation have been made to improve accuracy and reliability at extreme joint positions. Additionally, a transformation has been implemented that converts room coordinates to robot coordinates to account for the 45^o mounting angle of the arm. This facilitates operation of the robot by aligning the system coordinate frame to the operator's frame of reference. Efforts are now underway to reverse the conversion such that joint-to-point calculations can be made, enabling joint angles from either the robot itself or external sensors to be translated into spatial coordinates. This will be useful later for both feedback control of end-point and verification of intended position.

The serial interface to the robot has been updated to include a gripper command signal and eliminate any jitter of the arm due to movement command buffering. Communication is now bi-directional between the Robot Controller (RC) and the Master Controller (MC) to collect either joint angle or endpoint information from the robot.

Voice selection of on-screen labels has been tested and with user training (both of the operator and voice recognition system) is quite accurate. By using labels on the objects imaged and recognized by the REFES (RE) system, user voice commands promise to be an effective command source.

Future Work:

Testing of the MC serial interface with the RE system is the next step of the project. Before delivery of a full RE system, a small demonstration program is to be used to test receiving and interpretation of RE commands. A meeting is scheduled for the middle of July to finalize the remaining details of the interface between the CWRU and SIS portions of the final combined system. This meeting will also include any training necessary to set-up and operate the RE system upon delivery.

August 12, 2003 Progress Report

Progress:

The meeting between CWRU and SIS on July 11th was productive in helping to resolve a few of the remaining issues in coupling the two systems. Among these were the user interface, which was designed to present a simplified set of available commands and the structure of the serial interface. Other topics covered were computer requirements for the RE system and operation and calibration questions.

The interface between Master Controller and REFES has been created and preliminary testing looks favorable. A queue size of 30 trajectory points has been used for path planning and execution. A target date for delivery of the complete RE system has been tentatively set for the week of August 18th. A computer for running the RE software has been ordered, but a temporary machine will be used in the interim until it arrives.

A pneumatic end-effector (fig. 1a) has been built to grasp large cups such as thermal insulated coffee mugs (~70mm diameter, fig 1b). The choice of grasping relatively large objects was made for simplicity purposes and is a first step toward manipulating smaller objects later on. Additional fingers can easily be fabricated for the generic pneumatic actuator, depending on the size of objects to be picked up. The actuator is an SMC double acting pneumatic gripper with a 30^{o} range of motion. Maximum object size is about 100mm.

(a) (b)

Figure 1: Photographs of the end of arm tooling (a) and the cup it was designed to manipulate (b).

Future Work:

Final testing of the CWRU-RE interface will be conducted presently using the supplied test program. Upon set-up and installation of the RE system, evaluation will begin. Initial tests will include whole system spatial repeatability and information transfer rate. The feasibility of head orientation and voice input devices will also be explored.

September 4, 2003 Progress Report

Progress:

Continued testing of the RE and CWRU system interface has continued. The final issue seems to be a lack of recognition of replies sent back to the RE test program. Investigation with a different program (MS HyperTerminal) illustrates that the proper replies are indeed being sent. The problem is most likely a synchronization issue that should be resolved shortly.

Future Work:

Upon resolution of the above mentioned problem, a specific date will be set for delivery and installation of a complete RE system. After this time, complete testing of the system as a whole should progress rapidly.

September 22, 2003 Progress Report

Progress:

Communication issues between the REFES system and CWRU portion of the project have been resolved. Proper instructions and replies are now being sent successfully between each system. The RE system has been delivered and installed on-site at CWRU and preliminary testing looks good. The computer running the RE software, however, is deficient in memory and processor speed, resulting in fatal errors and termination of the RE program. A new, more capable system is currently being ordered to address this and should arrive in the next couple of weeks.

Future Work:

Upon receiving the proper hardware, testing should progress rapidly. As the system is evaluated and refined, additional levels of complexity will be added, including placing additional objects in the visual scene and smaller objects than those currently being manipulated.

October 3, 2003 Progress Report

Progress:

We are still waiting for the upgraded computer and have not begun any rigorous testing of the REFES system. Qualitative use looks good and user training with various input devices has begun to rule out learning effects on performance.

Future Work:

Current input devices to be investigated include voice command and two different types of head orientation sensors. Each of these takes the place of the mouse. User feedback regarding input method and ease of RE operation will be collected. Later testing will include placing additional objects in the visual scene as well as manipulating smaller objects.

November 14, 2003 Progress Report

Progress:

Significant progress has been made in the prior month. A new version of the RE software appears to overcome the memory buffer overrun issues seen in the previous version. This has allowed for thorough testing of the meshing of the RE and CWRU portions of the system. Several tests have shown that the CWRU system receives joint angle commands and passes them to the robot. Feed back of the actual joint positions from the robot confirms that the intended joint angles were reached.

The position specified by the RE system was off, however, by about a 1 inch error. This error is large enough to preclude grasping larger items such as the coffee mug used in testing. Images and data files from the RE system were passed along to SIS for correction of the lens distortion file and transformation matrix. It is thought that adjusting these two components will improve the vision system accuracy. The new distortion file has been placed into the RE system and validation tests performed with the data again passed to SIS for interpretation. We are waiting for confirmation of it's accuracy as well as a new transformation matrix from SIS.

Future Work:

With the expected updated transformation matrix and validated distortion correction, the next phase of the project can be started. This entails testing the performance of the vision system with a test object in 13 locations throughout the visual field (as per SIS supplied test protocol). Protocol has been reviewed by CWRU and looks to be valid. Robot may be used as a coordinate measuring tool to locate each test position.

December 9, 3003 Progress Report

Progress:

Independent testing of the REFES (RE) system was conducted by Ardiem Medical with results still pending. Testing of robot arm accuracy was not performed due to continued work on the RE to robot coordinate transformation matrix. Data formatting on the joint values sent from the robot has been implemented to ensure that the joints values are read in the proper order. Portions of legacy code have been removed enabling the robot to match the trajectory sent from the RE system. Comparison of actual joint values to joint commands from REFES illustrates this.

Future Work:

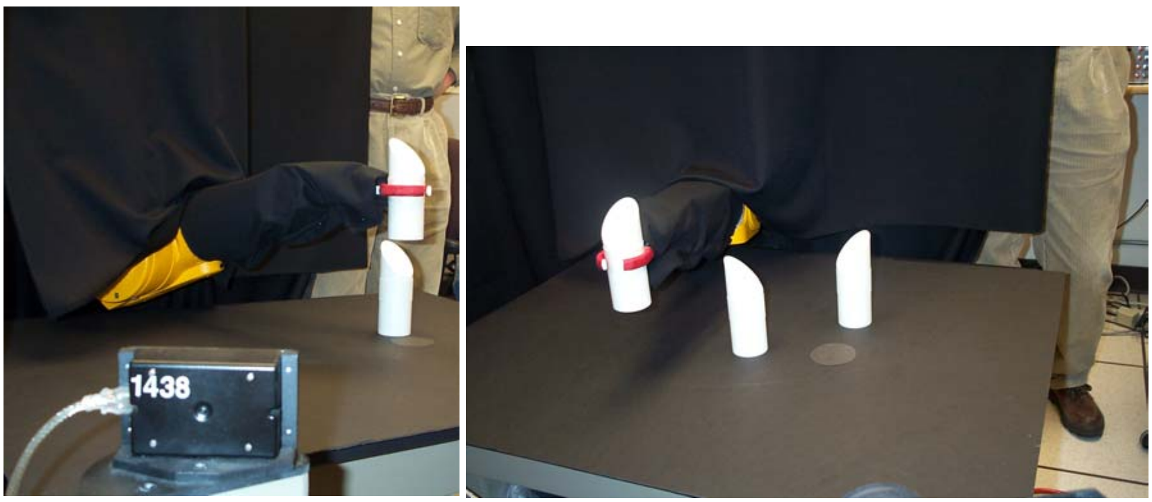
A new transformation matrix from SIS is still needed to complete phase 1 of the project. The next step includes implementation of end effector tracking with the two auxiliary cameras. Several additional commands need to be added to the CWRU portion of the system to facilitate this. Another visit to CWRU by both SIS and Ardiem is expected the week of December 15th. This will consists of delivery of the new RE tracking software as well as testing of robot accuracy.

January 14, 2004 Progress Report

Progress:

Significant progress has been made in the past few weeks concerning both the base system as well as the implementation of advanced features. At current, the whole system is nearly complete and almost fully functional.

The camera to robot coordinate transformation matrix has been calibrated such that the robot is now able to grasp and manipulate imaged objects (fig 1). Accuracy was also improved by re-calibrating the filter on the main camera flash such that the line pattern it projects is crisper and more vertical than before. Gripper to object error is now at most around 5mm, small enough to grasp the various objects used in this phase of the project.



(a) (b)

Figure 1: Photographs of a) Robot moving object and left tracking camera and b) robot moving object through "cluttered" workspace to test object avoidance.

The new version of the REFES software has been installed to implement the real-time tracking features. The system is now able to recognize the following abnormal conditions and audibly warn the user:

1) Misplaced objects (in the case of being knocked over or dropped)
2) New objects added to the workspace
3) Transient obstructions such as other people moving through the field of view

The tracking system is also able to monitor the position of the gripper, corroborating the data sent back from the robot. This feature is of particular interest, as potential later versions implemented in an FES application will require this information for closed loop position control. The frame rate is over 20Hz. Qualitative analysis of gripper tracking spatial accuracy is close to that of the main camera in some fields of view. It is thought that a larger "hand shaped" gripper with a more prominent profile will improve tracking accuracy in non-optimal fields of view.

Collision avoidance has also been added in this new version. The robot is able to move a grasped object over or behind other objects that may be in its path. This feature at current only applies to the grasped object, not the arm itself. Future version will need to address the issue of the robot arm colliding with items in the workspace.

The new RE software now includes the user options “Move object to mouth” and “Return object from mouth”. This, in addition to “fine control” of the destination point completes the basic user commands set out for the project.

Significant changes to the logic of the Master Controller and robot programs have also been made to accommodate new commands for the tracking features. Commands added to the MC/robot software include Pause and Resume as well as Get Joint Angle and Get Spatial Position to send the current robot state back to the RE system.

The robot now streams back to the MC both the current joint angles as well as current spatial position and orientation of the center of the gripper. The MC updates this information at 10Hz (down from 50Hz). The data format used between the MC and robot has been expanded to include a “hundreds” byte such that acceptable angle values now include from -180 to 999.99 degrees and spatial positions within the full work envelope of the robot can be transmitted more easily. A combination of reducing the update rate and error checking reduces the noise errors while permitting an increased amount of information to be passed between Mc and robot.

Qualitative assessment of the system has been performed, moving objects around on the work surface and to the “mouth” position. The robot is able to locate, grasp, and move known objects with a high degree of repeatability.

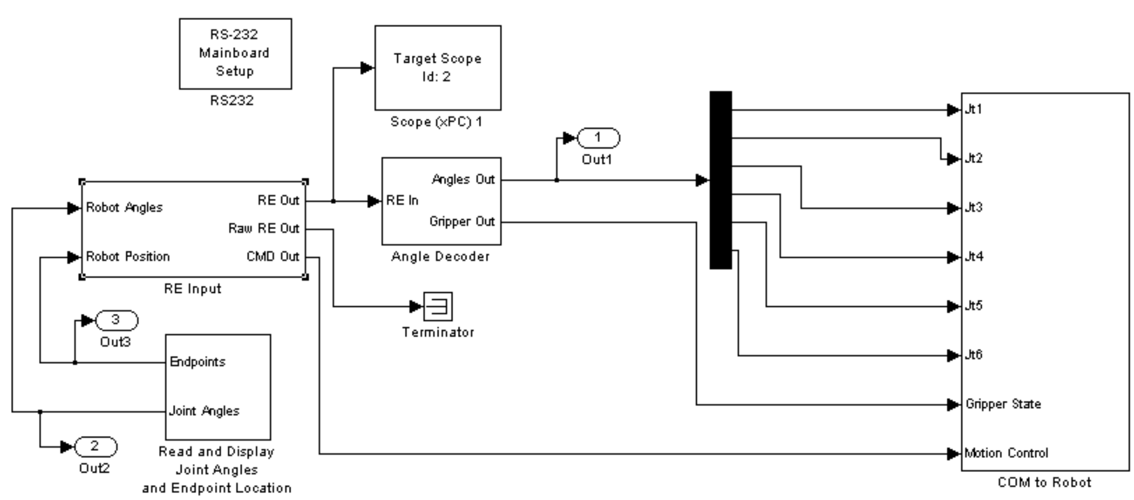
Voice recognition software has been tested with the RE user interface and works quite well. Training time for voice recognition is around 10 minutes for fairly accurate performance. The sparseness of the RE GUI is an important factor in this, but selection of novel destination points may require an inappropriate amount of time.

Future Work:

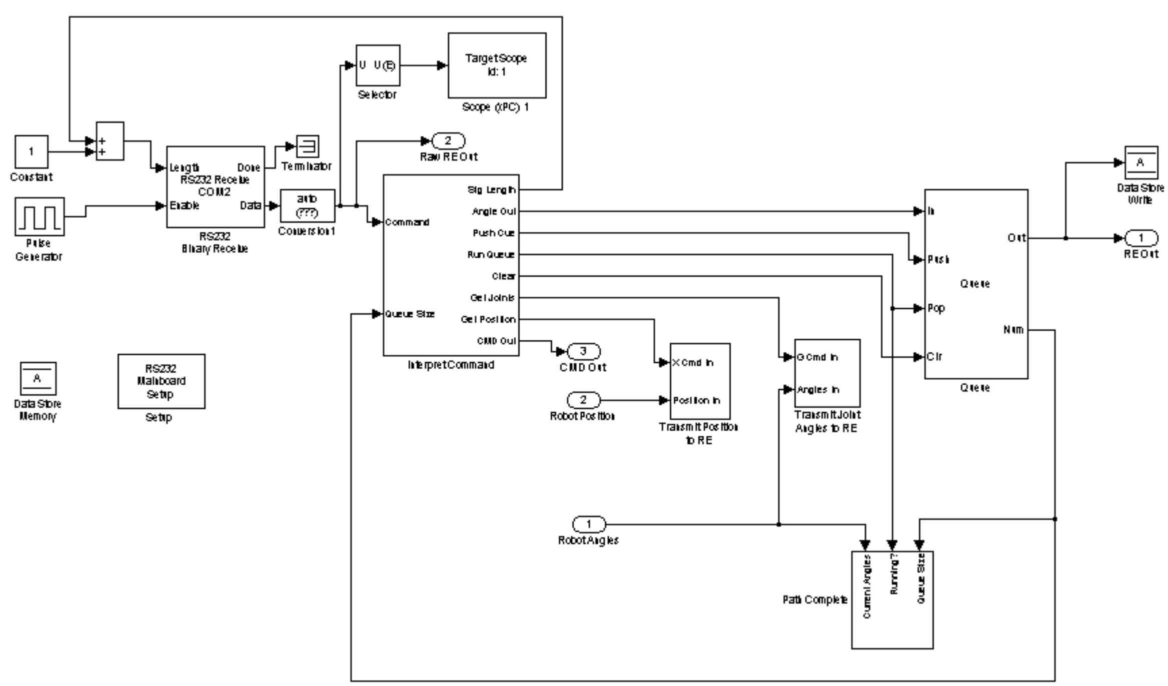
Little remains to be accomplished in this phase of the project. The most glaring issue to be solved is the trajectory used for the “Return from mouth” command. The RE system currently calculates an infeasible return path, crashing the robot. It is not known at this time if this is due to an internal logic error, or the result of bad information from the MC/robot. A final command, the Stop/Reset Queue command from REFES also needs to be implemented in the MC/robot software. Upon solution of these problems, quantitative functional tests of system performance will be performed. Subjects will be asked to select and move objects located in known positions to either other known positions or to and from the face area. The time required to select and manipulate the object will be recorded an evaluated as a measure of functional performance. Subjects will use both voice recognition and head-mouse inputs.

Appendix III: Simulink programming blocks used in the neuroprosthesis Master Controller:

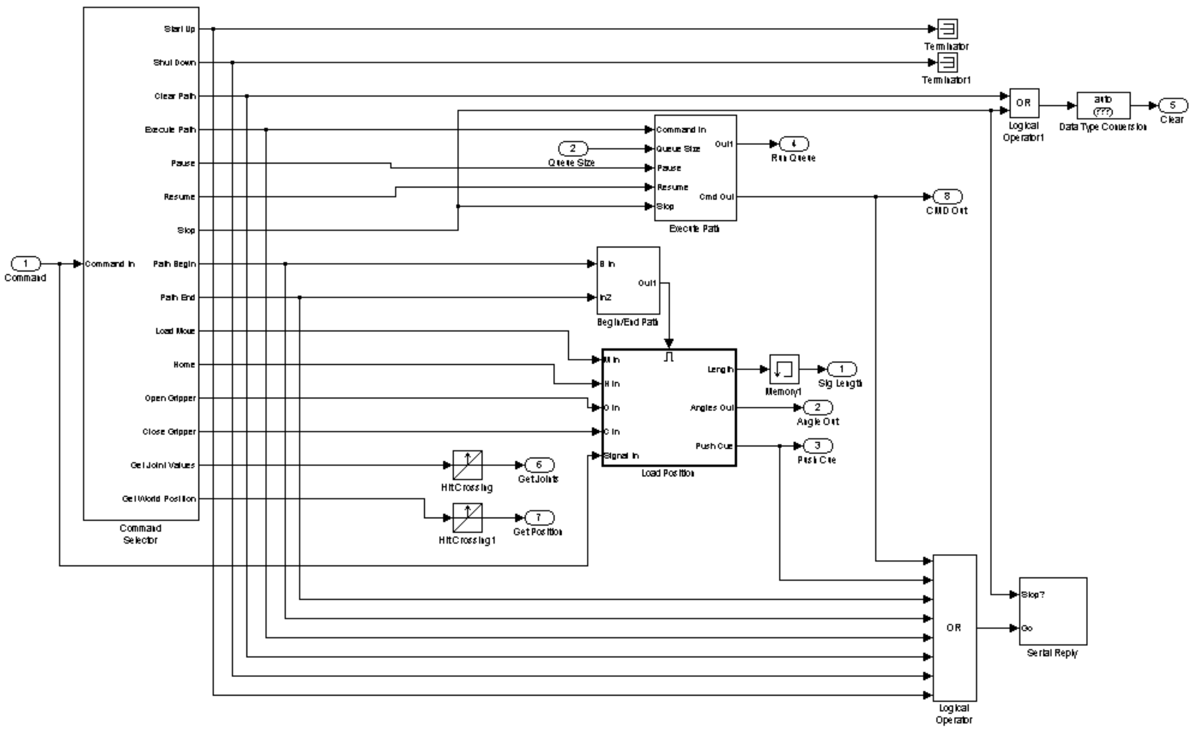
The following Simulink blocks were used in the software implementation of the Master Controller in this project. The first figure in this Appendix illustrates the main routine used for this Master Controller. The remaining figures illustrate blocks the are included either in the main block or in one of the sub-blocks. Note that these block diagrams are the equivalent of listing the software code for the Master Controller.



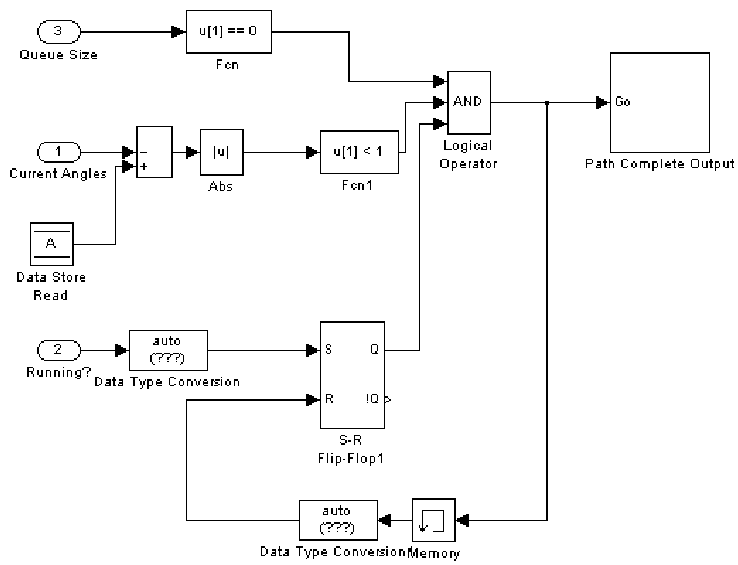
Main Simulink block



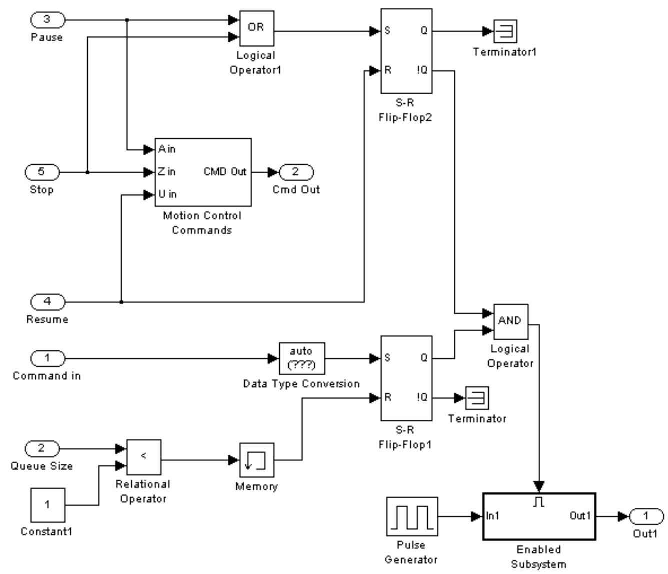
RE-input" Simulink sub-block



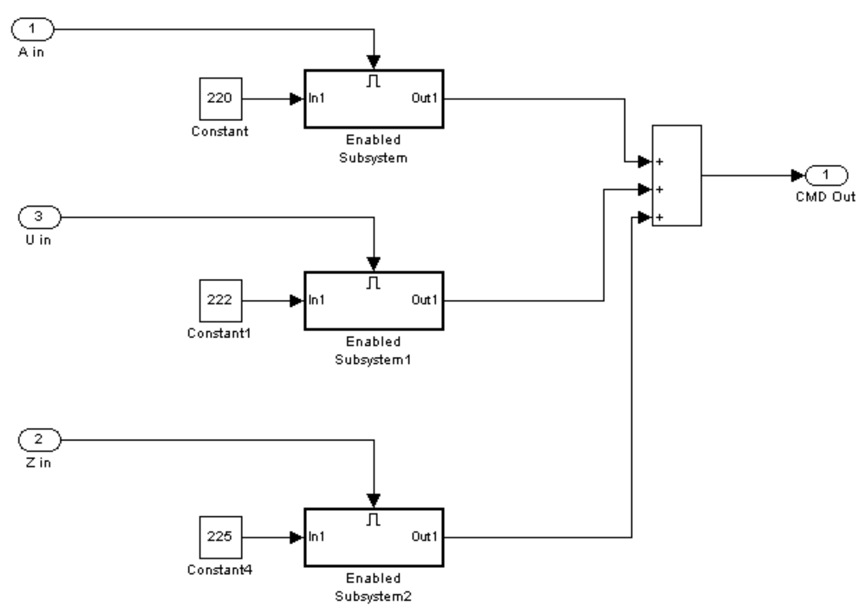
“RE Input/Interpret command” Simulink sub-block



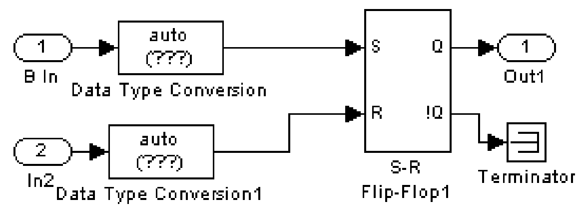
“RE Input/Path Complete” Simulink sub-block



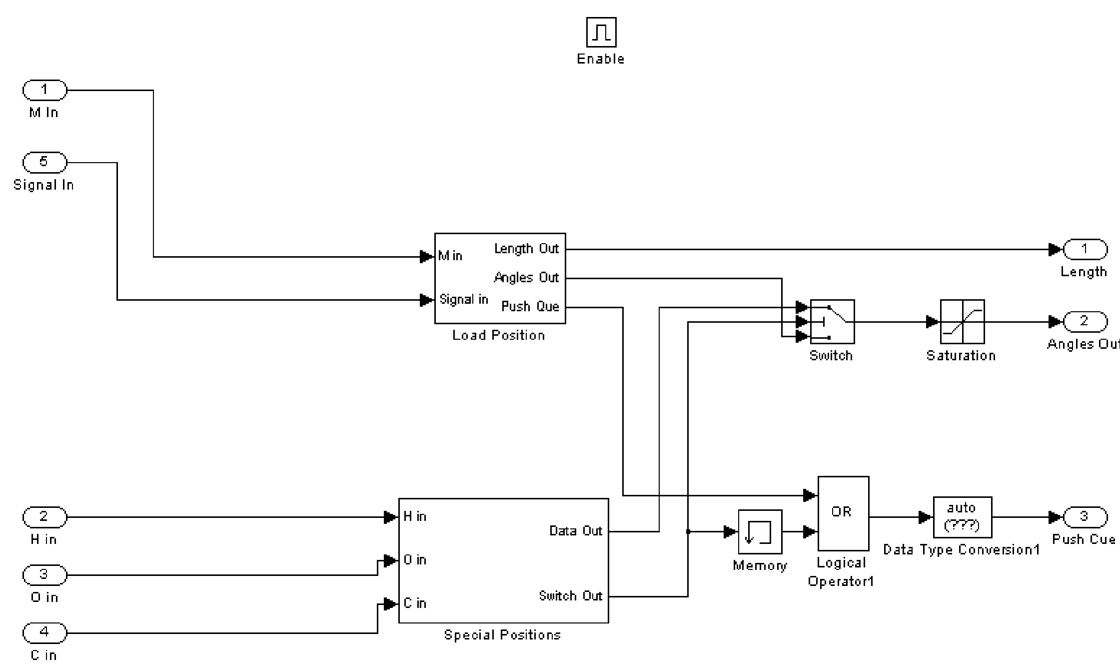
"RE Input/Interpret command/Execute Path" Simulink sub-block



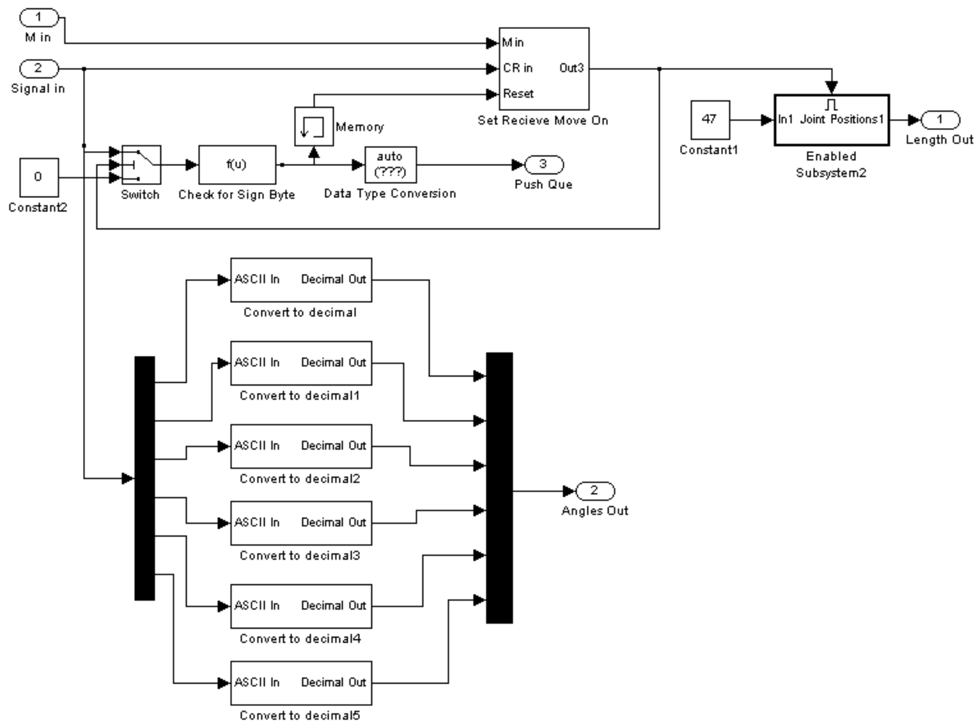
"RE Input/Interpret command/Execute Path/Motion Control Commands" Simulink sub-block



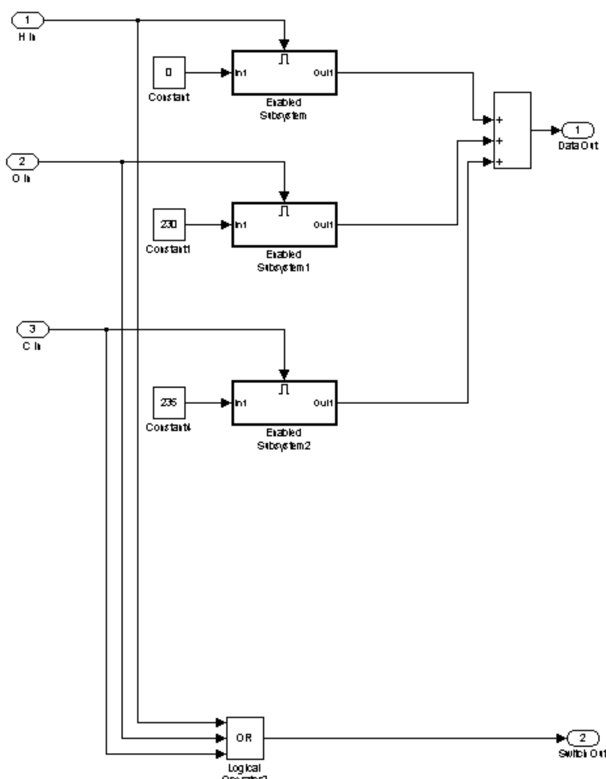
“RE Input/Interpret command/Begin/End Path” Simulink sub-block



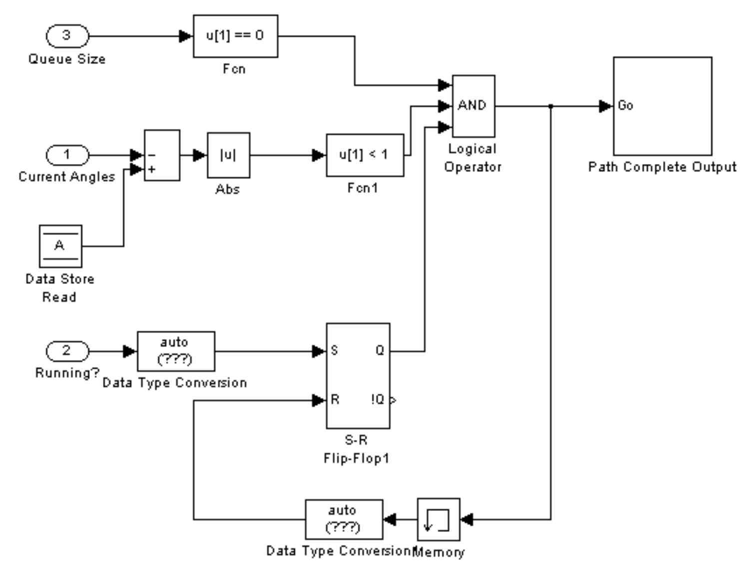
“RE Input/Interpret command/Load Position” Simulink sub-block



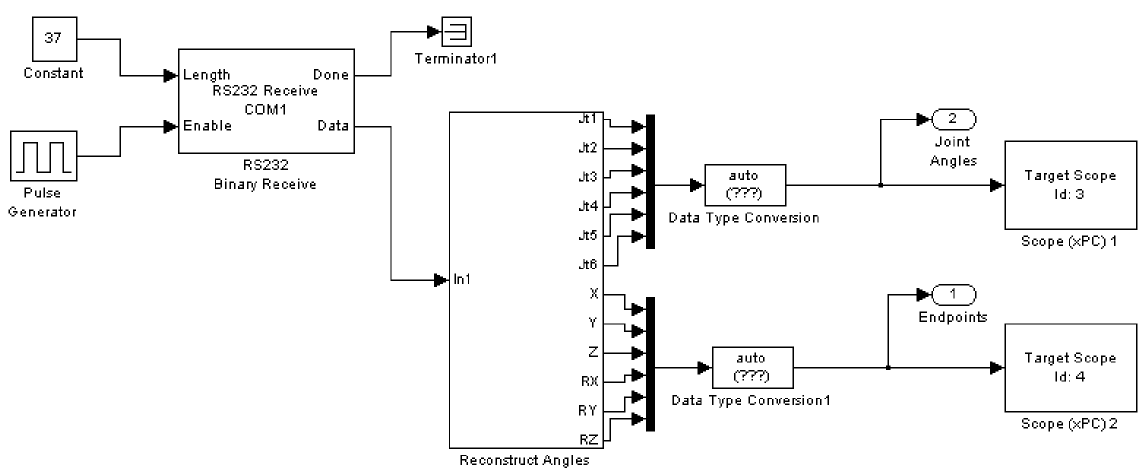
“RE Input/Interpret command/Load Position/Load Position” Simulink sub-block



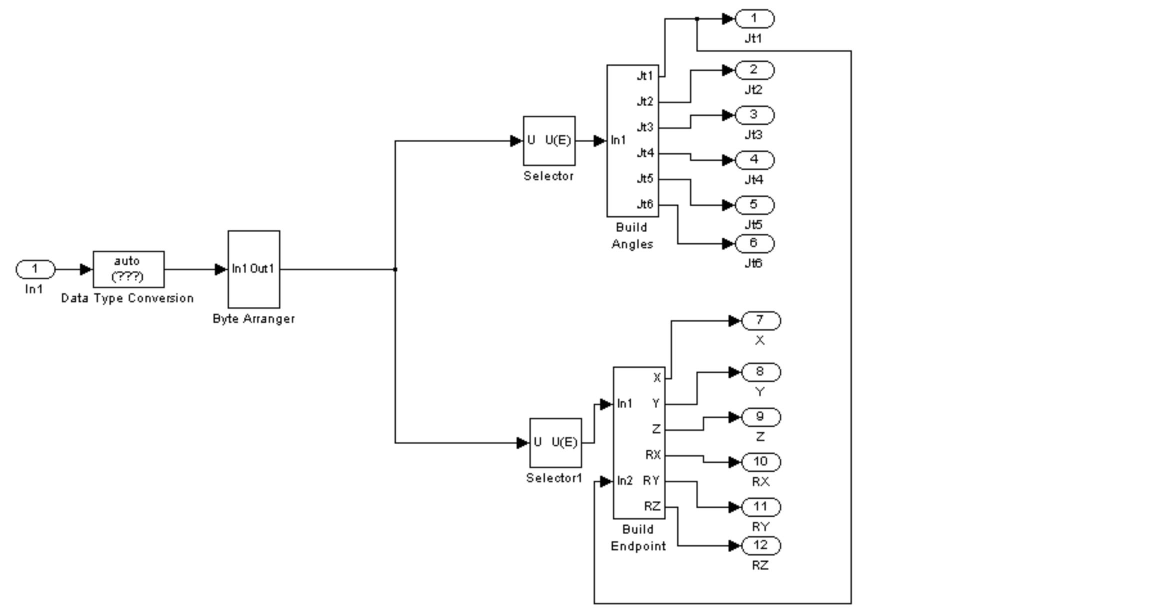
“RE Input/Interpret command/Load Position/Special Positions” Simulink sub-block



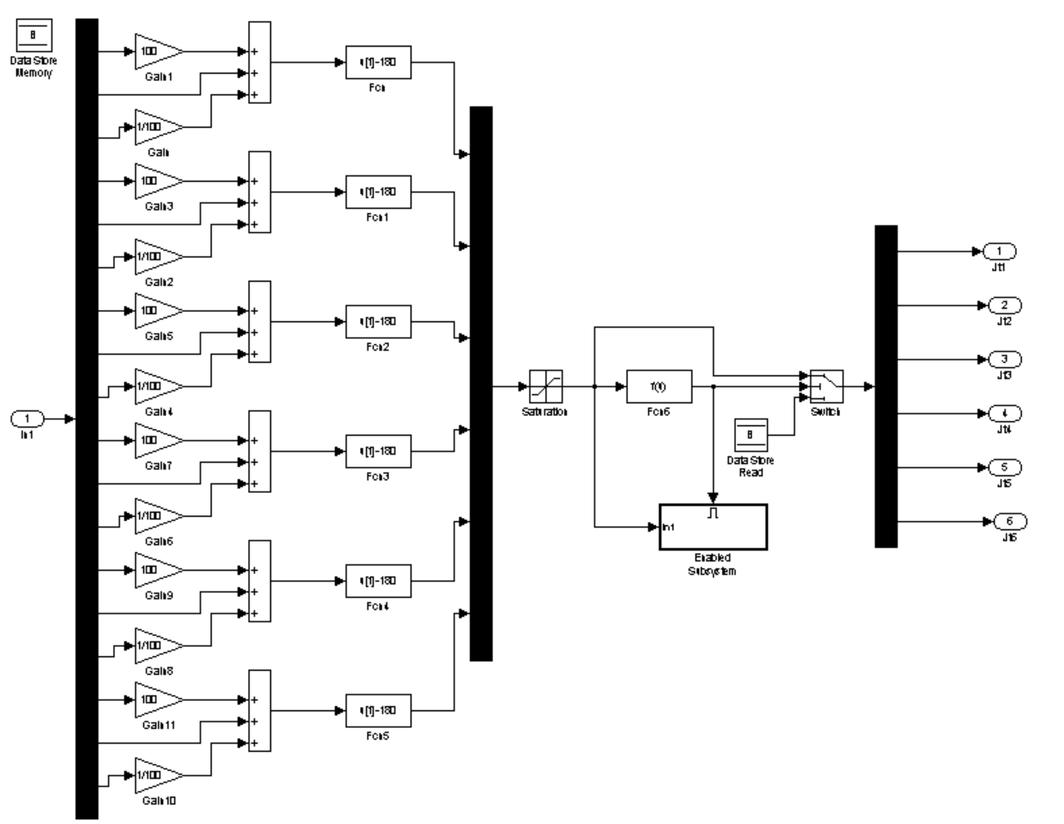
“RE Input/Path Complete” Simulink sub-block



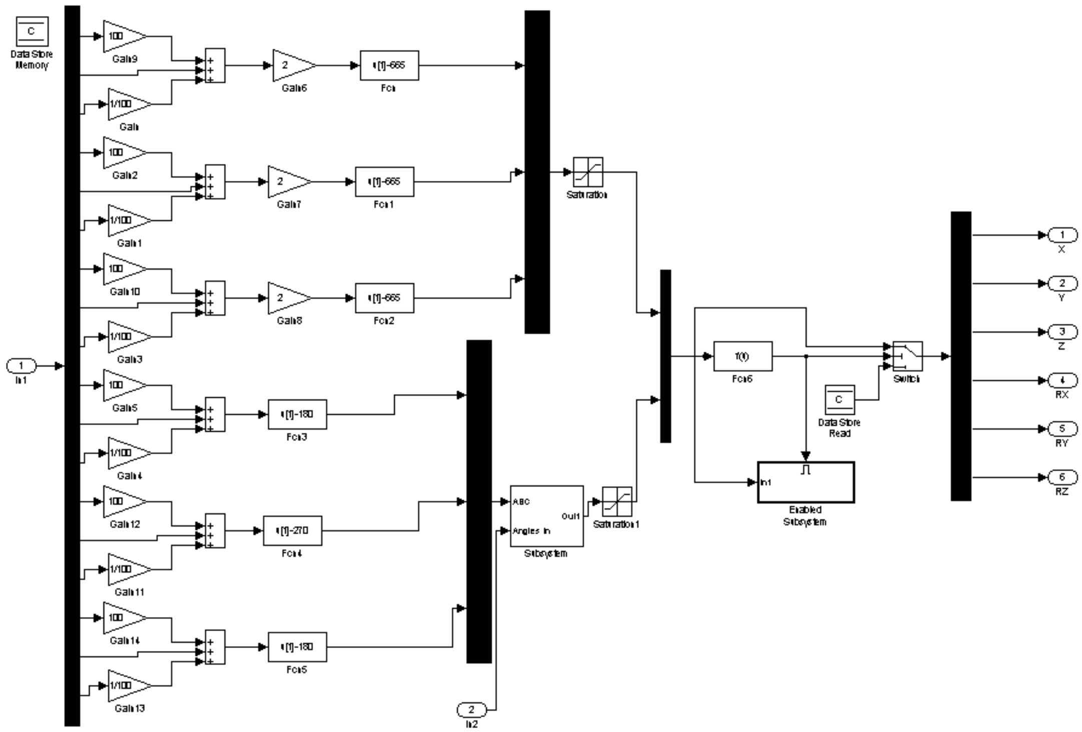
“RE Input/Read and Display Joint Angles and Endpoint Location” Simulink sub-block



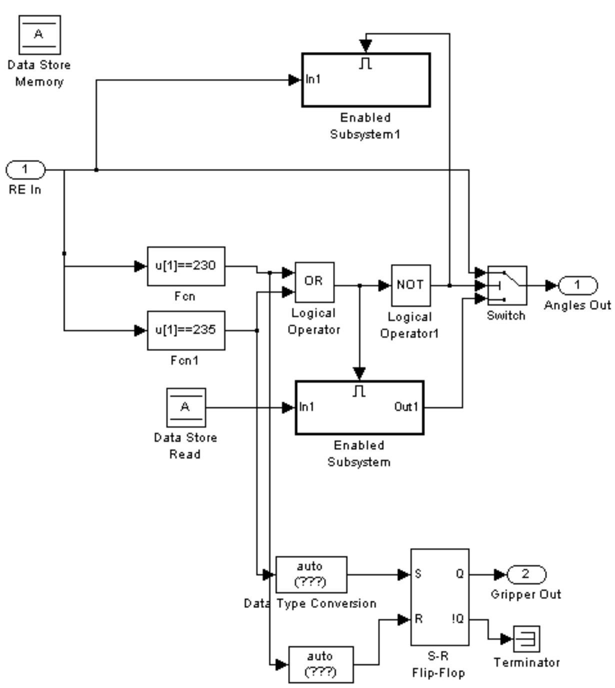
"RE Input/Read and Display Joint Angles and Endpoint Location/Reconstruct Angles" Simulink sub-block



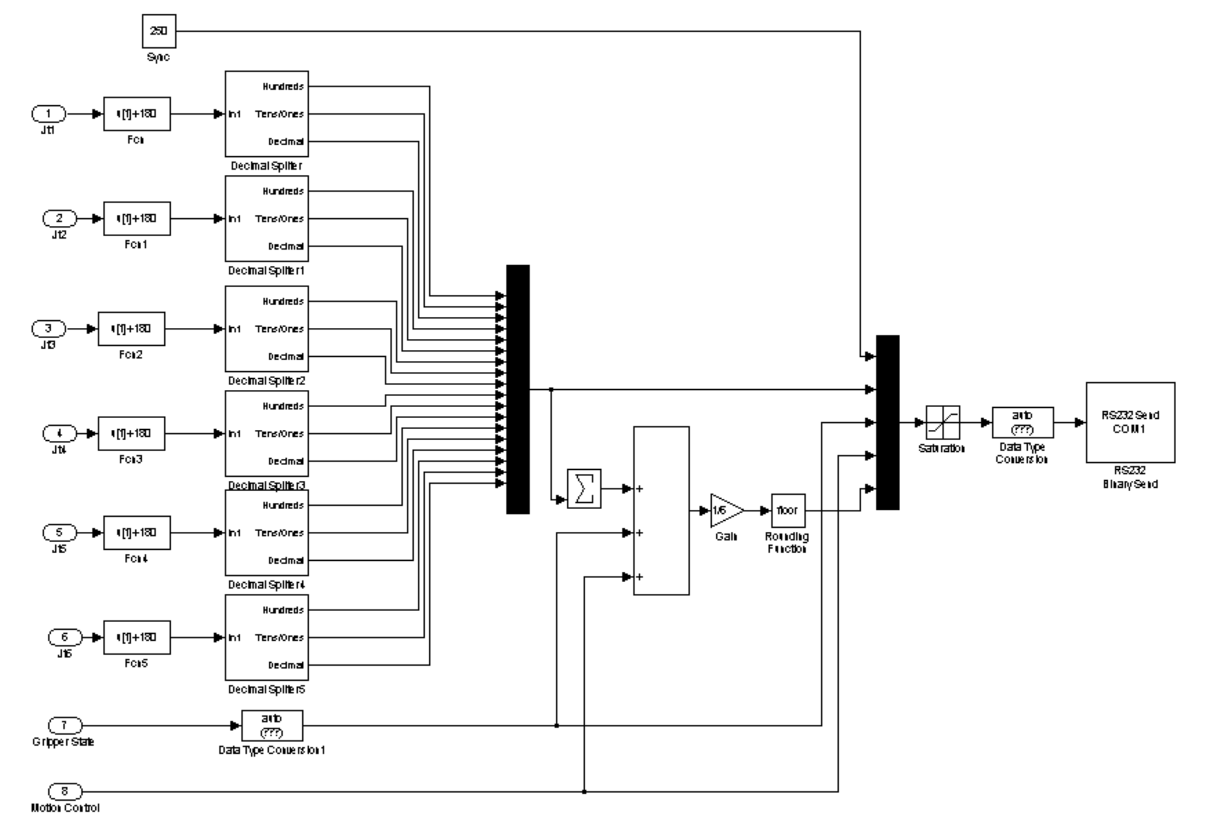
"RE Input/Read and Display Joint Angles and Endpoint Location/Reconstruct Angles/Build Angles" Simulink sub-block



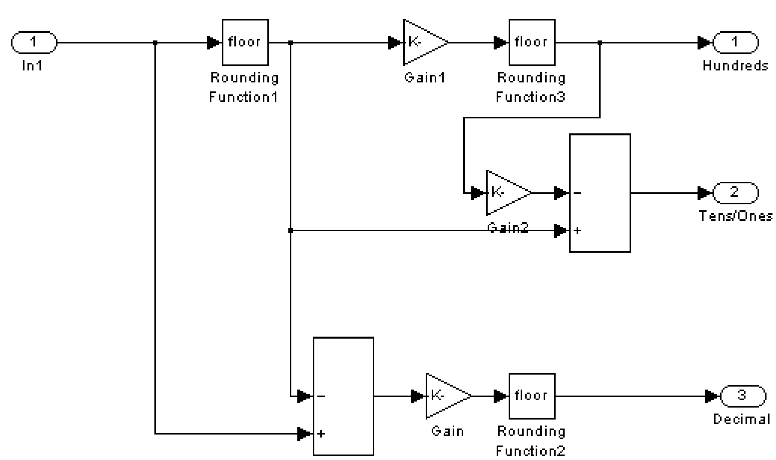
“RE Input/Read and Display Joint Angles and Endpoint Location/Reconstruct Angles/Build Endpoint” Simulink sub-block



“Angle Decoder” Simulink sub-block



“COM to Robot” Simulink sub-block



“COM to Robot/Decimal Splitter” Simulink sub-block

Final Report Appendix IV: References

Bach J, A. Zeelenberg, and C. Winter. Wheelchair-Mounted robot manipulators: Long term use by patients with Duchenne Muscular Dystrophy. *Am J Phys Med Rehabil*, vol. 69(2), pp. 55- 59, 1990.

Betz R., M. Mulcahey, B. Smith, R. Triolo, A. Weiss, M. Moynahan, M. Keith, and P. Peckham. Bipolar latissimus dorsi transposition and functional neuromuscular stimulation to restore elbow function in an individual with C4 quadriplegia and C5 denervation. *J. Amer. Paraplegia Soc*. 15:220-228, 1992.

Graupe D. An overview of the state of the art of noninvasive FES for independent ambulation by thoracic level paraplegics. Neurol Res. 24(5):431-42, 2002.

Hammel J, K. Hall, D. Lees, L. Leifer, M. Van der Loos, I. Perkash, and R. Crigler. Clinical evaluation of a desktop robotic assistant. *J Rehabil Research*, vol. 26(3), pp. 1- 16, 1989.

Hammel J, M. Van der Loos, and I. Perkash. Evaluation of a vocational robot with a quadriplegic employee. *Arch Phys Med Rehabil*, vol. 73, pp. 683-693, 1992.

Handa Y, T. Handa, and Y. Nakatsuchi. A voice controlled functional electrical stimulation system for the paralyzed hand. *Japanese Journal of Medical Electronics and Biological Engineering*, vol. 25, pp. 292-298, 1985.

Handa Y., A. Naito, M. Ichie, T. Handa, N. Matsushita, and N. Hoshimiya. EMG-Based Stimulation Patterns of FES for the Paralyzed Upper Extremities. *9th International Symposium on External Control of Human Extremities*. Dubrovnick. 329-340, 1987.

Handa Y, T. Handa, M. Ichie, H. Murakami, N. Hoshimiya, S. Ishikawa, K. Ohkubo. Functional electrical stimulation (FES) systems for the restoration of motor function of paralyzed muscles – versatile systems and a portable system. *Front Med Biol Engng*, vol. 4(4), pp. 241-255, 1992.

Hoshimiya N, A. Naito, M. Yajima and Y. Handa. A multichannel FES system for the restoration of motor functions in high spinal cord injury patients: A respiration-controlled system for multijoint upper extremity. *IEEE Trans Biomed Engr*, vol. 36(7), pp. 754-760, 1989.

Kameyama, J., Takahashi, H., Saito, C., Handa, T., Ichie, M., Handa, Y., Hoshimiya, N. Control of shoulder movement by FES (1). *Proceedings of the IEEE EMBS Conference* 12: 871-872, 1990.

Kameyama, J., Sakurai, M., Handa, Y., Handa, T., Takahashi, H., Hoshimiya, N. Control of shoulder movement by FES. *Proceedings of the IEEE EMBS Conference* 13: 871-872, 1991.

Kameyama, J., Handa, Y., Ichie, M., Hoshimiya, N., and Sakurai, M. Control of shoulder movement by FES. *15th Annual International Conference IEEE Engineering in Medicine and Biology Society*, 15: 1342-1343, 1993.

Kobetic R, Triolo RJ, Marsolais EB. Muscle selection and walking performance of multichannel FES systems for ambulation in paraplegia. IEEE Trans Rehab Eng. 5(1): 23-29, 1997.

Kralj AR, Bajd T. Functional Electrical Stimulation: Standing and Walking After Spinal Cord Injury. CRC Press Inc., pp 123-138, 1989

Lathem, P.A., Gregorio, T.L., Garber, S.L. High Level Quadriplegia: The Occupational Therapy Challenge. *American Journal of Occupational Therapy*. 39(11). 705-714, 1985.

Lauer R, P. Peckham, and K. Kilgore. EEG-based control of a hand grasp neuroprosthesis. *NeuroReport*, vol. 8(10), pp. 1767-1771, 1999.

Malick, M. and Meyer, C. "Manual on Management of the Quadriplegic Upper Extremity." Washington: Library of Congress. 1978.

Nathan, R.H. An FNS-Based System for Generating Upper Limb Function in the C4 Quadriplegic. *Medical & Biological Engineering & Computing.* 27. 549-556, 1989.

Nathan R and A. Ohry. Upper limb functions regained in quadriplegia: A hybrid computerized neuromuscular stimulation system. *Arch Phys Med Rehabil*, vol. 71, pp. 415-421, 1990.

Peckham, P. H. , Keith, M. W. "Motor prostheses for restoration of upper extremity function." In Neural Prostheses: Replacing Motor Function After Disease of Injury, eds. R.B. Stein, P.H. Peckham, and D.P. Popovic. New York: Oxford University Press, 1992.

Prochazka A, M. Gauthier, M. Wieler, Z. Kanwell. The Bionic Glove: An Electrical Stimulator Garment that Provides Controlled Grasp and Hand Opening in Quadriplegia. *Arch Phys Med Rehabil* 78:608-614, 1997.

Regalbuto M, T. Krouskop, and J. Cheatham. Toward a practical mobile robotic aid system for people with severe physical disabilities. *J Rehabil Research*, vol. 29(1), pp. 19-26, 1992a.

Regalbuto M, T. Krouskop, and J. Cheatham. Applying robotic technology to aid people with severe disabilities. *Assist Technol*, vol. 4, pp. 87-94, 1992b.

Seamone W and G. Schmeisser. Early clinical evaluation of a robot arm/worktable system for spinal cord injured persons. *J Rehabil Research*, vol. 22 (1), pp. 38-57, 1985.

Triolo R, C. Bieri, J. Uhlir, R. Kobetic, A. Scheiner, and E. Marsolais. Implanted FNS systems for assisted standing and transfers for individuals with incomplete cervical spinal cord injuries. *Arch Phys Med & Rehab,* 77:1119-1128, 1996.

Appendix VI – REFES Testing Final Report

REFES Testing
Final Report

Report Date
February 12, 2004

by
James R. Cupp

Abstract
The purpose of this report is to summarize the series of tests performed on the REFES system on Government Prime Contract Number N00014-02-C-0384. Two identical REFES systems were interfaced with two commercially available fully articulated robotic arms at two research locations. The REFES systems as interfaced with the robotic arms were tested for accuracy, repeatability, object targeting, obstacle avoidance, and the ability to track the motion of the robotic arm in a real time or close to real time manner. The results of the tests indicate that the REFES system may be an acceptable method of controlling a functional electrical stimulation device.

Table of Contents

Tables

Appendices

Summary

The primary goal of the testing program for the REFES System was to evaluate the system for potential use with a Functional Electrical Stimulation system or a prosthetic arm or other prosthetic device. A series of tests were performed on two identical REFES Systems that were interfaced with two different commercially available fully articulated robotic arms at two separate research locations. Each system was tested for accuracy, repeatability, object targeting, obstacle avoidance, and the ability to accurately track the motion of the robotic arm in a real time or close to real time manner as the arm moved an object across the robot working-table. Data that was collected characterized the capabilities of the system and provided information for future improvement of the system. The accuracy and repeatability tests yielded an average of an 8.68-millimeter error value for the worst-case axis and a 1.6-millimeter overall average repeatability error. Object targeting was directly related to the accuracy testing results and was within the error values obtained in the accuracy tests. Obstacle avoidance testing was software driven and was overall successful within the limits of the robot work envelope. Real time robot arm tracking accurately monitored the robot arm coordinates but exhibited a constantly increasing coordinate error at extreme distances from the VZX imaging camera. The constant trend observed during the testing was that the coordinate error values were very location specific on the robot working-table but repeatable within a specific location. This trend can be minimized and the overall accuracy of the system increased by improvement of the transformation matrix accuracy to compensate for the observed error pattern. Hardware changes such as different lens systems may also be considered to reduce distortion effects that may be a cause of the observed error pattern. The general conclusion and recommendation indicated from this series of tests is that the REFES System as tested is satisfactory to be applied as a control and feedback device for Functional Electrical Stimulation, prosthetic limbs or other assistive devices.

Introduction

The purpose of this report is to summarize the series of tests performed on the REFES System on Government Prime Contract Number N00014-02-C-0384 for Spatial Integrated Systems, Inc.. Identical REFES systems were interfaced with two different commercially available fully articulated robotic arms at two research centers. The REFES systems as interfaced with the robotic arms were tested for accuracy, repeatability, object targeting, obstacle avoidance, and the ability to track the motion of the robotic arm in a real time or close to real time manner. Each of the individual tests that were performed is briefly described in the following text along with a summary of the test results and the resulting conclusions.

Test Summaries

Test One

Initial Accuracy Test of the REFES System with the Mitsubishi Robot

Purpose

The purpose of this test series was to challenge the ability of the REFES System interfaced with a Mitsubishi RV1A robot to accurately measure the three dimensional location and orientation of an object within the workspace and to accurately transform that data to coordinates usable by the robotic arm.

Test One - Test Procedure Summary

The REFES system interfaced with a Mitsubishi RV1A robotic arm and attached to a work platform was used in this test series. A description of the physical set up of the system is described in Appendix A. The robot working-table was divided into ten areas that evenly covered the semicircular reach of the robot arm. (See fig.1 for an illustration of the robot working-table). A hexahedral object was placed on the robot working-table at each of the locations shown in figure 1and multiple iterations of positional and rotational orientations were performed at each of the locations. The actual x and y- axis coordinates and orientations of the object at each individual location were physically measured from ground fiduciary surfaces on the robot base using precision mechanical measuring tools. The data from the recorded physical measurements and the REFES system calculated coordinates were compared to yield error values for each of the locations on the robot working-table to produce an error map of the robot working-table.

Test One - Results Summary

The results of this initial test yielded inconsistent coordinates for each of the locations on the robot working-table (See Test One – Table 1). The x and y-axis error values were very high at some of the locations and iterations. The standard deviation was extremely high as well indicating extreme variability. The rotational orientation (Yaw) of the object was also subject to a high degree of variability yielding high error values. The coordinates that exhibited the most consistent data and yielded very low error values were the z-axis, pitch, and roll angular offset.

Test One – Table 1

Overall Average Composite Error of All Locations				
	X	**Y**	**Z**	**Yaw**
	Mm	**Mm**	**mm**	**deg**
Average Error	25.3	8.9	-1.5	-1.7
Standard Deviation	22.5	20.1	6.4	15.9
Maximum Value	101.7	64.1	11.6	50.6
Minimum Value	-38.2	-38.5	-11.2	-41.9

Test One – Conclusions Summary

The average error values are deceptively low as evidenced by the large standard deviations and large minimum and maximum values for each coordinate (See Test One – Table1). The errors values tended to be location specific but were extremely inconsistent and subject to large differences between sequential data points that simply modifying the transformation matrix to correct the errors would not be possible. The yaw or rotation angle variability seemed to be partially caused by the hexahedral test object lacking unique features that would assist with orientation. The yaw angle at sixty degrees in some cases was misinterpreted as a minus thirty-degree angle. The overall results were poor enough to warrant the consideration of a repeat of this test using a different test object that would be more conductive to proper interpretation by the REFES system.

Test Two
Accuracy Test of the REFES System with the Mitsubishi Robot – Retest

Purpose

The purpose of this test series was to repeat the Initial Accuracy Test with a more appropriate and well-defined test object. The goal of this test was the same as the previous test which was to challenge the ability of the REFES System interfaced with a Mitsubishi RV1A robot to accurately measure the three dimensional location and orientation of an object within the workspace and to accurately transform that data to coordinates usable by the robotic arm.

Test Two - Test Procedure Summary

The REFES system interfaced with a Mitsubishi RV1A robotic arm and attached to a work platform was used in this test series as with the previous accuracy test. A description of the system is described in Appendix A. The robot working-table was divided into ten locations that evenly covered the semicircular reach of the robot arm. (See fig.1 for illustration of the robot working-table). The previous accuracy test indicated that coordinate errors were location specific on the robot working-table. This test series was truncated to use six of the ten locations marked on the robot working-table. Each of the locations chosen corresponded to the extremes of the robot working-table and would accurately map the coordinate error values for the robot working-table. For these tests the locations one, three, five, six, eight, and ten were used. A cylindrical object 58 mm in diameter and 160 mm in height with an oblique angle cut at the top of the cylinder from the 160 mm height projecting toward the base at a forty-five degree angle was used as the test object for this set of tests (See fig 3 for a drawing of the test object). The test object was placed on one of the locations to be tested and coordinate data was taken using the REFES system. Data was taken at each location at five different yaw-axis angular rotations. Coordinate data was taken for each location and rotation a total of three iterations each. The actual x and y- axis coordinates and orientations of the object at each individual location were physically measured from ground fiduciary surfaces on the robot base using precision mechanical measuring tools. The data from the recorded physical measurements and the REFES system calculated coordinates were compared to yield error values for each of the locations on the robot working-table to produce an error map of the robot working-table.

Test Two - Results Summary

The results of this second accuracy test yielded very consistent coordinates for each of the locations on the robot working-table (See Test Two – Table 1). The x and y-axis error values were considerably lower than those in the first set of tests (Compare with Test One- Table1).

The yaw axis variability also saw a significant reduction in variability. The z-axis, pitch, and roll angular error values remained a low value as was observed in the first accuracy test.

Test Two – Table 1

Overall Average Composite Error of All Location Data				
	X	**Y**	**Z**	**Yaw**
	mm	**Mm**	**mm**	**Deg**
Average Error	-8.68	0.52	-3.31	-1.44
Standard Deviation	4.03	3.64	1.63	2.64
Minimum Value	-15.3	-4.7	-8.6	-8.9
Maximum Value	-3.0	6.4	-0.4	4.7

Test Two - Conclusions Summary

The error values shown in the data chart above is a composite of all of the locations added together and is misleading in some respects (See chart Test Two – Table 1). The standard deviation and difference between the minimum and maximum value spread for individual locations is actually much less than shown above. The error value for each location is very consistent regardless of yaw axis rotation. The coordinate error does continue to produce a very location specific error value that seems to be related to the distance from and the angular deflection of the object from the centerline of the REFES system (VZX Imaging System) camera. The coordinate error values for each location and as a summed whole were with in an acceptable and very repeatable error range for functional use of the robot arm with these coordinates. The REFES System coordinate data and the transformation matrix that was used to convert the REFES coordinates to the robot coordinates were very successful in this set of tests. The repeatable and location specific coordinate error values indicate potentially greater accuracy is available with this system. The REFES system as was tested produces more than sufficient accuracy for the intended purpose of the system.

Test Three
Accuracy Test of the REFES System with the Staubli Robot

Purpose

The purpose of this test was the same as the accuracy tests performed using the REFES System interfaced with a Mitsubishi RV1A robot. This set of tests will utilize the REFES System interfaced with a Staubli robotic arm. The goal of this test was to accurately measure the three dimensional location and orientation of an object within the workspace and to accurately transform that data to coordinates usable by the Staubli robotic arm.

Test Three - Test Procedure Summary

The REFES system was interfaced with a Staubli robotic arm that was suspended above a worktable identical to the worktable used in the REFES system - Mitsubishi robot accuracy tests. A description of the physical set up of the system is described in Appendix B. The robot working-table was divided into twelve locations that evenly covered the active reach range of the robot arm. (See fig.2 for illustration of the robot working-table). Each of the locations to be tested for accuracy were distributed across the active robot working-table to evaluate the error values for the extremes and intermediate locations of the robot working-table. The same cylindrical test object was used as the test object for this set of tests as in the second set of accuracy tests with the Mitsubishi robot (See fig 3 for a drawing of the test object). As with the previous accuracy tests the test object was placed on one of the locations to be tested and coordinate data was taken using the REFES system (VZX Imaging System). Data was taken at each location at five different yaw-axis angular rotations. Coordinate data was taken for each location and rotation for a total of three iterations at each location. The physical x and y- axis coordinates of the locations on the robot working-table were not able to be measured using mechanical measuring tools. The Staubli robot did not have any mechanical fiduciary reference surfaces from which to measure coordinates as were available on the Mitsubishi robot. The x and y-axis coordinates of the test locations were obtained by gripping the test object with the end effector claw on the robot arm and moving the test object until the test object was centered on the test locations physically drawn on the robot working-table and then reading the coordinate data from the robot controller display. The data from the recorded robot coordinate measurements and the REFES system calculated coordinates were compared to yield error values for each of the locations on the robot working-table to produce an error map of the robot working-table. Location number one (See figure Fig. 2) was outside of the imaging field of the VZX Imaging System Camera and was excluded from the testing procedure.

Test Three - Results Summary
The results of the accuracy test with the REFES system interfaced with the Staubli robot were overall very poor (See Test Three –Table 1). The x and y-axis coordinate error value averages appeared relatively normal but the standard deviation and the minimum and maximum error values revealed extreme variations. The extreme variations were primarily caused by anomalous data from three single iterations in three different locations and orientations. Data from other locations and orientations in the same test set exhibited normal and consistent error values. In the data charts listed below the overall average composite error values are shown with the anomalous data (Test Three – Table 1) and with the anomalous data not included (Test Three-Table 2). Without the anomalous data the z and yaw axes exhibit normally low error values. The x and y axis averages are within normal values but some extreme error values exist for some locations as illustrated by the high standard deviation and the high minimum and maximum values.

<u>Test Three – Table 1</u>

Overall Average Composite Error of All Location Data

	X Mm	Y mm	Z mm	Yaw deg
Average Error	8.87	-10.39	0.88	-1.99
Standard Deviation	47.59	45.82	23.38	1.11
Minimum Value	-497.05	-182.44	-4.95	-4
Maximum Value	117.84	518.1	298.3	-0.06

Test Three - Table 2

Overall Average Composite Error of All Location Data Less Anomalous Readings

	X Mm	Y mm	Z mm	Yaw deg
Average Error	11.33	-12.65	-0.98	-1.98
Standard Deviation	25.25	14.65	2.02	1.12
Minimum Value	-32.08	-47.94	-4.95	-4.00
Maximum Value	65.68	14.22	3.01	-0.06

Test Three - Conclusions Summary

The Staubli robot set up is significantly different than the Mitsubishi robot set up. The Staubli robot is suspended above the robot working-table and utilizes a claw-like scissors type of end effector. The Mitsubishi robot is mounted to the robot working-table and utilizes a parallel leg end effector (See photograph 1 for Mitsubishi robot). The utilization of the claw end effector as a fiduciary reference object for the REFES system was hypothesized as possibly being more difficult to obtain accurate reference coordinates than the end effector on the Mitsubishi robot and may be the main source of error. The error values for the x and y-axis in several locations appeared to have a weak correlation between yaw angle but was not consistent enough to trend.

The extremely high anomalous data contained in three iterations was not explainable nor were they repeatable. The elimination of the anomalous data iterations from the averaged data yielded error values that were location specific as seen in previous accuracy tests and seem to be related to the distance from and the angular deflection of the object from the centerline of the REFES system (VZX Imaging System) camera. After excluding the three high anomalous data iterations the overall error values still exhibit a high degree of error values for the x and y-axis especially for some locations. The overall accuracy of this system is marginally functional for further testing. Further testing may be completed if the transformed data compensates for the errors at each location or if the testing is limited to locations that exhibit the lowest error values.

Test Four
Targeted Object Test and Inert Object Avoidance Test of the REFES System with the Staubli Robot

Purpose

This test series consisted of two separate sets of tests. The first test set was to determine the ability of the REFES system to locate targeted objects on the robot working-table. The second test set was to move a target object between two points by navigating past inert objects (obstacles) between the origin and destination locations.

Test Four - Test Procedure Summary

The REFES system was interfaced with a Staubli robotic arm that was suspended above a worktable identical to the worktable used in the REFES system - Mitsubishi robot accuracy tests. A description of the system is described in Appendix B.

Targeted Object Test

Six of the twelve locations within the active reach range of the robot arm that exhibited the lowest error values were selected for the targeted object test. (See fig.2 for illustration of the robot working-table). The locations selected were locations four-A, five, six, seven eight, and nine. The x and y-axis coordinates of the test locations were obtained by gripping the test object with the end effector claw on the robot arm and moving the test object until the test object was centered on the test locations physically drawn on the robot working-table and then reading the coordinate data from the robot controller display. A test object (See figure 3) was placed on one of the test locations. The object coordinates were scanned by the REFES system and the object location was displayed graphically on the computer monitor relative to the robot working-table. The object was selected for the robot to move to the object and grasp the object. As the object was grasped the offset caused by the gripping of the object from the target location on the robot working-table was measured physically and recorded.

Inert Object (obstacle) Avoidance Test - Single Object Avoidance

The robot in this test is to grasp and move the test object from an origin location to a target location and avoid an inert object that had been placed in the path between the two locations. Four locations were selected for the single object avoidance test that allowed an adequate distance of between the location and a target location to allow an inert object to be inserted in the path. The four locations selected were four-A, five, seven, and nine (See figure two for locations).

Inert Object (obstacle) Avoidance Test - Multiple Object Avoidance

A second variation with the Inert Object Avoidance Test was to place multiple objects (up to three) in the path between an origin location and a target location. The origin location for each of these tests was location nine since this location offered the greatest distance from the target destination location to adequately place multiple inert objects.

Test Four - Results Summary

Test Four – Table 1

Targeted Object Test

Offset at Grasp By Robot

Location	X mm	Y mm
9	2	15
8	8	8
7	-3	-2
6	4	17
5	2	12
4A	0	-2
Average	2	8

Test Four – Table 2

Inert Object Avoidance Test

Origin Location	Destination Location	Results
9	Target	Avoided object
7	Target	Avoided object
5	Target	First attempt hit obstacle, Second attempt succeeded
4A	Target	Avoided object

- Multiple Obstacle Avoidance

All paths are from location 9 to target

Iteration	Object to Avoid	Results
Trial 1	Two cylindrical objects to avoid	Objects avoided
Trial 2	One cylindrical and one Lego block set	Objects avoided
Trial 3	One cylindrical object, one gray cup, and one Lego block set	Gray cup not recognized and knocked over
Trial 4	One cylindrical object, one short six sided object, and one Lego set	Objects avoided

Trial 5	One cylindrical object, one gray cup, and one Lego block set	Objects avoided, gray cup location since Trial three was changed and recognized by system

Test Four - Conclusions Summary
The coordinates obtained by the REFES system and transformed into robot coordinates seems adequate as evidenced by the successful location and grasping of the test object. The offset (error) values adequately reflect equal or better values than those obtained from the prior initial accuracy test. Offset values obtained in this test may have been compromised by the type and lack of precision of the end effector used on the robot.
The inert object avoidance test was generally successful with two exceptions that may have been caused by limitations in the robot envelope or the software.
The object avoidance tests were to be repeated using the REFES system and Mitsubishi robot at Spatial Integrated Systems. The REFES system - Mitsubishi robot exhibits a greater degree of accuracy and has been optimized as the primary test bed for this project. Results from the Mitsubishi robot will yield a more adequate representation of the capabilities of the REFES system.

Test Five
Inert Object Avoidance Test of the REFES System with the Mitsubishi Robot

Purpose
The purpose of this test series was to determine the ability of the REFES system to detect an inert object placed in the workspace and to alter the path of the active moving object to avoid the inert object. The REFES system is expected to monitor the workspace and prompt the user to reconstruct the 3D environment coordinates if a new object or objects are detected prior to movement of the active object by the robot.

Test Five - Test Procedure Summary

The REFES system interfaced with a Mitsubishi RV1A robotic arm and attached to a work platform was used in this test series. A description of the system is described in Appendix A.

A spherical test object approximately 2.75” in diameter was placed on the robot working-table within the reach of the robot arm. (See photo 1). After a scan by the REFES system the object was moved to a new location during which the robot coordinates were periodically recorded to create an electronic file of the object path. The object was returned to the origin point and an inert object (obstacle) was placed in the path between the origin and destination locations. After a second scan by the REFES system to obtain the coordinates of the inert object the robot was commanded to move the test object from the origin location to the destination location (See Photo 2). The robot coordinates were periodically recorded to create an electronic file of the new path as the robot path was altered to avoid the inert object. The above test sequence was repeated at several locations on the robot working-table. The robot trajectory paths with and without an obstacle were graphically plotted (See Test Five - Graph Set One) for each set of locations. The success or failure of the robot to avoid the inert object was qualitatively recorded as well. An abbreviated set of trials was also performed using multiple objects.

Test Five - Results Summary
See Graph Set One for path with and path without obstacle.

Test Five – Table 1

Single Obstacle Avoidance Tests	
Iteration	**Results**
Test One	Successfully avoided obstacle
Test Two	Successfully avoided obstacle
Test Three	Successfully avoided obstacle
Test Four	Successfully avoided obstacle
Test Five	Successfully avoided obstacle
Test Six	Successfully avoided obstacle
Test Seven	Successfully avoided obstacle
Test Eight	Successfully avoided obstacle
Test Nine	Successfully avoided obstacle
Test Ten	Successfully avoided obstacle

Test Five – Table 2

Multiple Object Avoidance Test	
Iteration	**Results**
Test Eleven	Successfully avoided obstacles
Test Twelve	Successfully avoided obstacles
Test Thirteen	Successfully avoided obstacles

Conclusions Summary
The REFES system – Mitsubishi robot system was found to be able to identify and successfully alter the path of the robot when moving an object to avoid obstacles on the robot working-table (See Test Five Tables 1 and 2).

The REFES system is able to identify when an object is placed on the robot working-table or in the path of the object being moved by the robot and to prompt the user by use of an alarm and by halting the robot operation.

Some limitations of the system that can be corrected by software changes mainly deal with limitations of the robot work envelope. Objects that are too close to the edge of the robot work envelope or penetrate the work envelope in the z-axis can in some cases be a cause of error in object avoidance.

Some limitations are hardware related. Objects that are shadowed or are too close to each other tend to be blended together can also cause an error in object avoidance.
The set of limitations described above have been seen in previous tests and were avoided when performing this set of tests.

Test Six
Final Test and Validation of the REFES System as a Feedback Control for Functional Electrical Stimulation - Object Tracking Test

Purpose
The purpose of this test is to determine the ability of the REFES system to act as a feedback control for Functional Electrical Stimulation by tracking the robot arm and test object as it is being moved between locations on the robot working-table.

Test Six - Test Procedure Summary
The REFES system interfaced with a Mitsubishi RV1A robotic arm and attached to a work platform was used in this test series. A description of the system is described in Appendix A. Position coordinates were recorded simultaneously from the both the robot controller and from the stationary motion monitoring cameras as the robot arm moved a test object between various locations on the robot working-table (See Fig.1). The difference in value between corresponding data points in the two sets of coordinates yielded an error value for each axis and set of locations.

Test Six - Results Summary
The resulting coordinate data from the robot and motion monitoring (tracking) cameras were entered into an Excel spreadsheet and analyzed for differentials corresponding data points that would yield an overall error value for each test iteration. The two sets of averaged data shown below as Iteration One and Iteration Two represents the results of robot object movement paths that covered the extreme range of the robot working-table. The complete set of data for both Iteration One and Iteration Two is illustrated graphically in a plot of the robot and tracking camera coordinates in Test Six - X-Y Graph 1.

Test Six – Table 1

Object Tracking Data Iteration One

	X-Axis Robot - Tracking Differential	Y-Axis Robot - Tracking Differential	Z-Axis Robot - Tracking Differential
Average Error	-7.44	-10.596	4.894
Standard Deviation	14.472	7.828	7.31
Minimum Value	-32.988	-27.27	-4.922
Maximum Value	13.144	1.896	22.568

Test Six – Table 2

Object Tracking Data Iteration Two

	X-Axis Robot - Tracking Differential	Y-Axis Robot - Tracking Differential	Z-Axis Robot - Tracking Differential
Average Error	-3.459	-9.125	4.749
Standard Deviation	15.718	9.249	7.327
Minimum Value	-31.359	-23.818	-9.109
Maximum Value	19.131	12.534	20.809

Test Six - Conclusions Summary

Two types of error factors are of concern in this test. The first factor is that the previous accuracy testing with this system indicates a consistent and repeatable error value of up to twelve millimeters in some locations and some axes and may be a factor in the robot trajectory data. The second factor is that the tracking camera errors, if any, may be additive, subtractive, or of no effect on the errors that the robot trajectory may include. At the time of this test accuracy data from the stationary tracking camera system was not available.

The x and y-axis coordinates exhibited the greatest error between the robot coordinates and the tracking camera coordinates from the center to the left side of the robot working-table as viewed from the perspective of the midpoint between the two stationary tracking cameras and facing the robot. The z-axis coordinates exhibited the greatest error between the robot coordinates and the tracking camera coordinates as the robot arm was at the home position or was traveling to or from the home position.

Both the x and y-axis coordinates error values tended to increase with distance from the REFES system camera. This error trend was the most evident at the left side of the robot working-table but was noticeable to a lesser degree on the right side of the robot working-table as viewed from the perspective of the midpoint between the two stationary tracking cameras and facing the robot. Some of the error values have been hypothesized to be a result of the progressive distortion effects of the camera lens proportional to increasing angular displacement from the centerline of the camera lens. The results tend to support this hypothesis although further investigation is warranted.

REFES Test Series Conclusions

One of the primary goals of the REFES System was to qualify the system for use with a Functional Electrical Stimulation system or a prosthetic arm or other prosthetic device. The various tests that were summarized in this report were designed to test the accuracy, repeatability, object targeting, obstacle avoidance, robot working environment monitoring, and the ability to track the motion of the robotic arm in a real time or close to real time manner. The REFES System was interfaced with two different commercially available fully articulated robotic arms at two locations. At the Spatial Integrated Systems facility a Mitsubishi RV1A robotic arm was interfaced with The REFES System and at Case Western Reserve University a Staubli robotic arm was interfaced with the REFES System. Both robots were fully articulated six axes of motion robots. The Mitsubishi robot was mounted with the base mounted to the test surface and the Staubli robot was mounted inverted to a pedestal to more closely mimic normal physiologic operation of a human arm. Development of the REFES System was accomplished first with the Mitsubishi robotic arm and when optimized then transferred to the Staubli robotic arm.

As a result the REFES System Mitsubishi robotic arm system was more advanced on the optimization curve than the Staubli robot system. Test results and conclusions in this section will deal primarily with the REFES System as interfaced with the Mitsubishi robotic arm.

Accuracy of the REFES System relies on coordinate data obtained using VZX Imaging System and relative to the VZX imaging System camera and using a transformation matrix algorithm to change the coordinate origin to that of the robotic arm. In the accuracy tests that were performed the coordinates errors between the actual robot coordinates and the transformed REFES System coordinates were well within the accuracy that would be required by using a prosthetic limb. Error values of all locations averaged less than nine millimeters and the maximum error at a single location did not exceed sixteen millimeters. The average repeatability within a specific location yielded error values of less than two millimeters. In conversations with personnel at Case Western a common accuracy to be expected by a person using a prosthetic limb or Functional Electrical Stimulation system was approximately twenty-five millimeters. The error values for each location revealed a trend for the errors to increase with increasing distance from the VZX Imaging System cameras and for angular displacement from the centerline of the camera. The transformation algorithm could be modified to improve the accuracy of the system beyond the current results.

Targeting testing was related to the accuracy testing and was performed without any difficulties being encountered. The accuracy of the targeting of an object was within the error values for each of the target locations. As long as the proper coordinates are inputted to the robot the robot has more than enough of an accuracy tolerance to perform the targeting operation.

Object avoidance testing was performed with only a few difficulties not related to the actual system. In some instances overly large test obstacles that penetrated outside of the working envelope of the robot were not processed correctly to create an avoidance path. Also objects that were shadowed or were too close to another object were viewed as a single object and were treated as such in creating an avoidance path. Other than the limitations mentioned, each of the tests performed moved test objects between locations on the robot working-table and successfully avoided single or multiple obstacles placed in the path of the moving test object. The robot path was normally to avoid an obstacle by moving over the obstacle vertically but in some cases the path went around an object.

Monitoring of the robot working-table environment by the RobotEyes system consistently and successfully detected changes in the robot work environment when the robot was at rest. When the robot is not moving and a new object or other change is detected in the robot working-table environment, the RobotEyes system triggers an alarm so that the user can make the decision on whether or not to rescan the workspace with the VZX Imaging system to reconstruct the new 3D environment.

Monitoring of the robot path in the robot working-table environment by the RobotEyes system consistently and successfully detected obstacles that were placed in the path of the moving object when the robot was in the process of moving an object from one location to another on the robot working-table. When the robot is moving, only its moving path is monitored for any new object and all other areas of the robot working-table are ignored. Once a new object is detected in the moving path of the robot, the robot will stop immediately. If this object moves out of the robot path within two seconds, the robot will continue its originally planned path. Otherwise, an alarm will sound and the user will be prompted to either "ignore" the new object, or "stop" current move, or "continue" the move. If the user chooses "ignore", the robot will move along its originally planed path regardless of the newly detected object. If "stop", the robot will put the object being moved on the table and go back to its home position. If "continue", the robot will find a new path using the estimated coordinates from the monitoring cameras to avoid any collision with the newly detected object and move to its destination point.

Feedback control or object tracking was performed by comparing the coordinates of the robot arm as the arm moved an object from one location to another on the robot working-table with the calculated coordinates as obtained from two stationary tracking cameras. The results of the testing indicated the robot and tracking camera coordinates correlating very well overall with no erratic coordinate tracking by the tracking cameras. As observed in previous accuracy testing the differences in coordinates between the robot and stationary tracking cameras followed a definite pattern. The further the object was from the VZX Imaging System camera the greater the difference between the two coordinates was observed. The error can be easily seen in the X-Y graphs of the travel of an object across the extreme limits of the robot working-table. The error values in the graph indicate a smooth curve that should be able to be compensated for by modifying the transformation matrix algorithm.

The general conclusion of the testing performed with the REFES System interfaced with a Mitsubishi RV1A robot indicates that the REFES System is sufficiently accurate, functional, and appropriate to be interfaced with a with a prosthetic arm or limb, functional electrical stimulation device, other prosthetic or assistive device. The accuracy and performance level of the system as tested is currently satisfactory for the intended purpose. The performance envelope of the REFES System can be improved by modest changes in the transformation matrix algorithms used to convert the visually obtained coordinates to a mechanical reference coordinate source. Hardware changes such as different lens systems may also be considered to reduce image distortion effects that may be a cause of the observed error pattern.

References

Test Report Initial Accuracy Test of RobotEyes™ System; Test Date, August 4-6, 2003; Report Date September 3, 2003 by James R. Cupp

Test Report Accuracy Test of RobotEyes™ System - Test Two; Test Date November 20, 2003; Report Date December 8, 2003 by James R. Cupp

Test Report Accuracy Test of RobotEyes™ System with the Staubli Robotic Arm; Test Date, December 3, 2003; Report Date January 23, 2003 by James R. Cupp

Test Report Targeted Object and Inert Object Avoidance Test of the RobotEyes™ System with the Staubli Robotic Arm; Test Date December 28, 2003; Report Date January 30, 2004 by James R. Cupp

Test Report RobotEyes™ – SIS Mitsubishi Robot Inert Object Placement Test and Inert Object Random Placement Test; Test Date January 13, 2004; Report Date February 7, 2004 by James R. Cupp

Test Report Final Test and Validation of the RobotEyes™ System as a Feedback Control for Functional Electrical Stimulation; Test Date January 29, 2004; Report Date February 10, 2004 by James R. Cupp

FIGURE 1
RELATIVE LOCATIONS OF TEST LOCATIONS – MITSUBISHI ROBOT

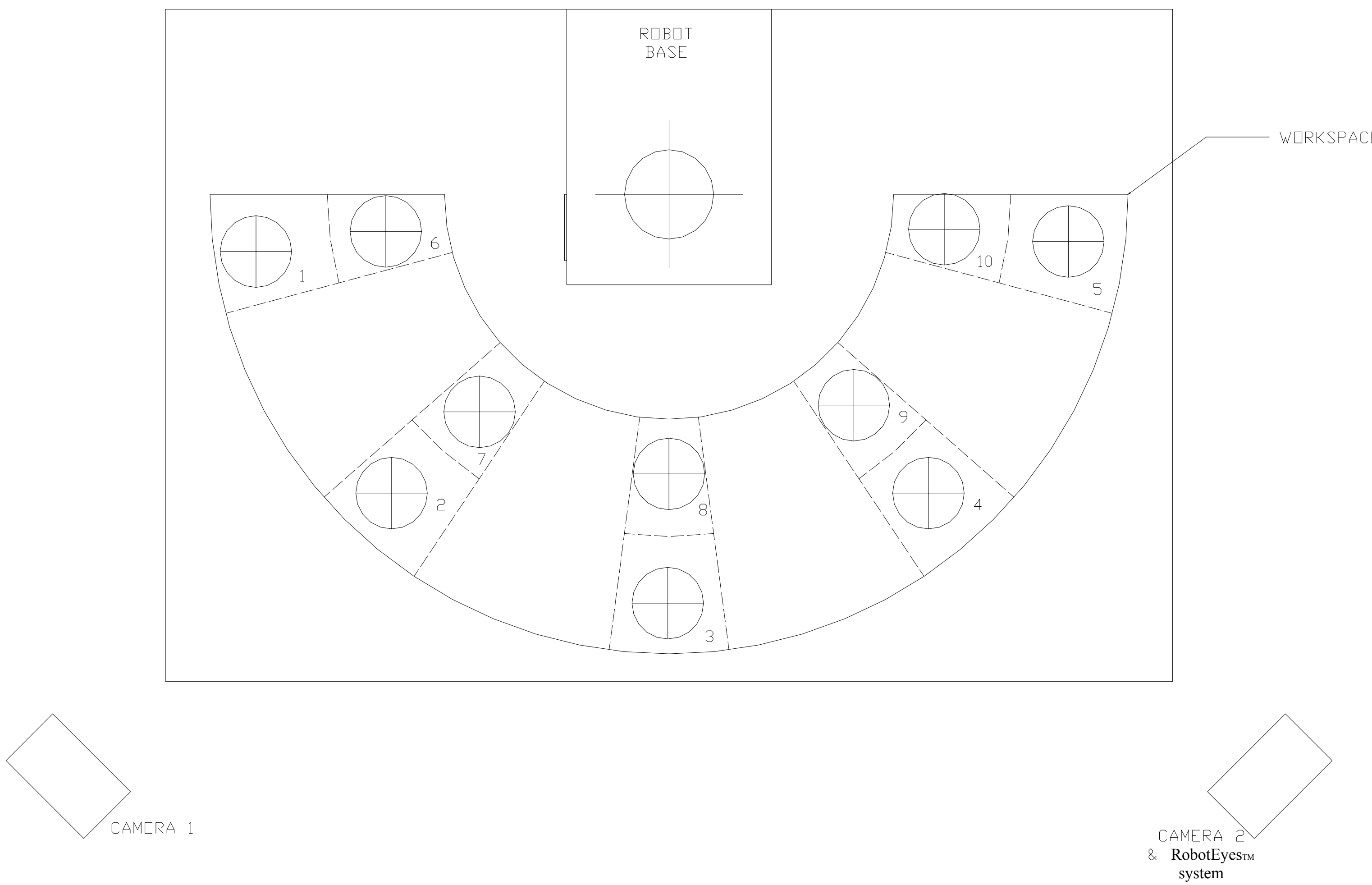


FIGURE 2
RELATIVE LOCATIONS OF TEST LOCATIONS - STAUBLI ROBOT

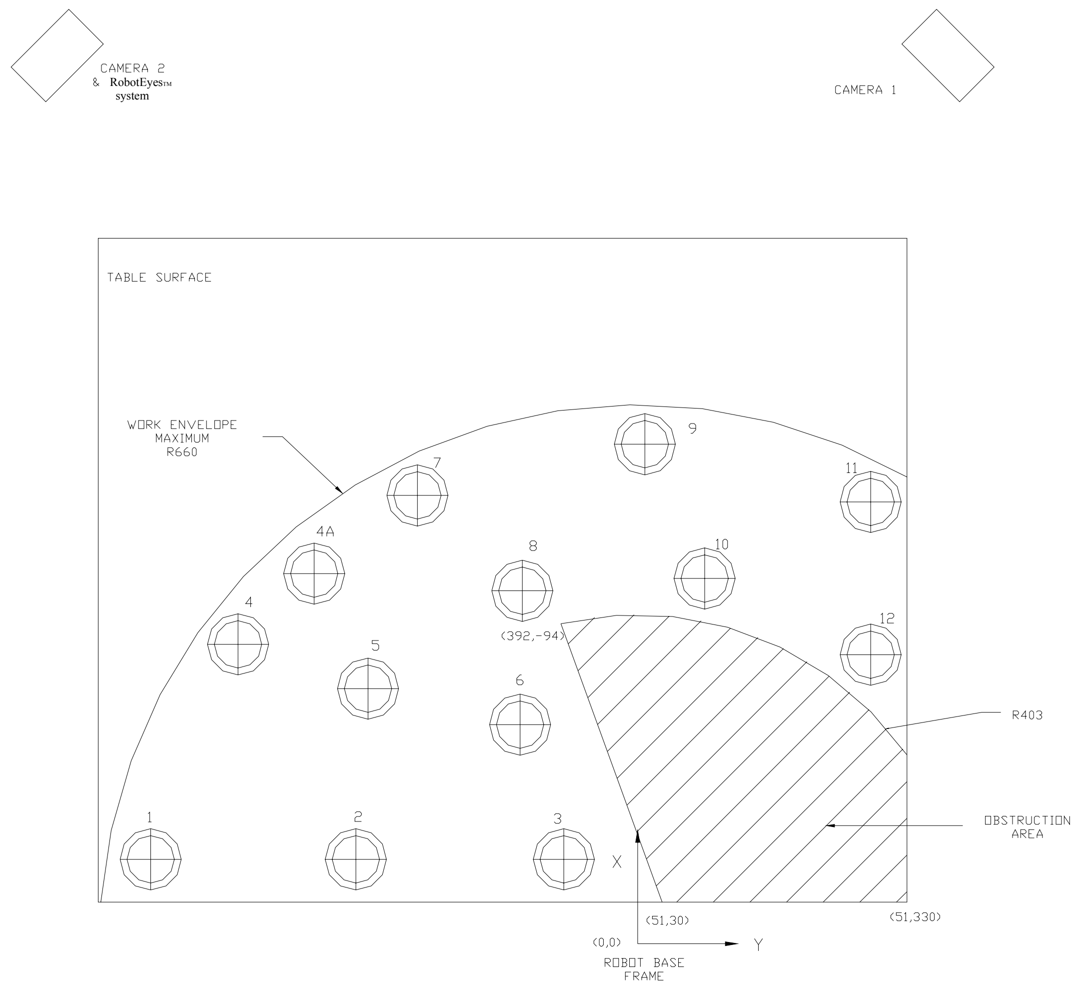
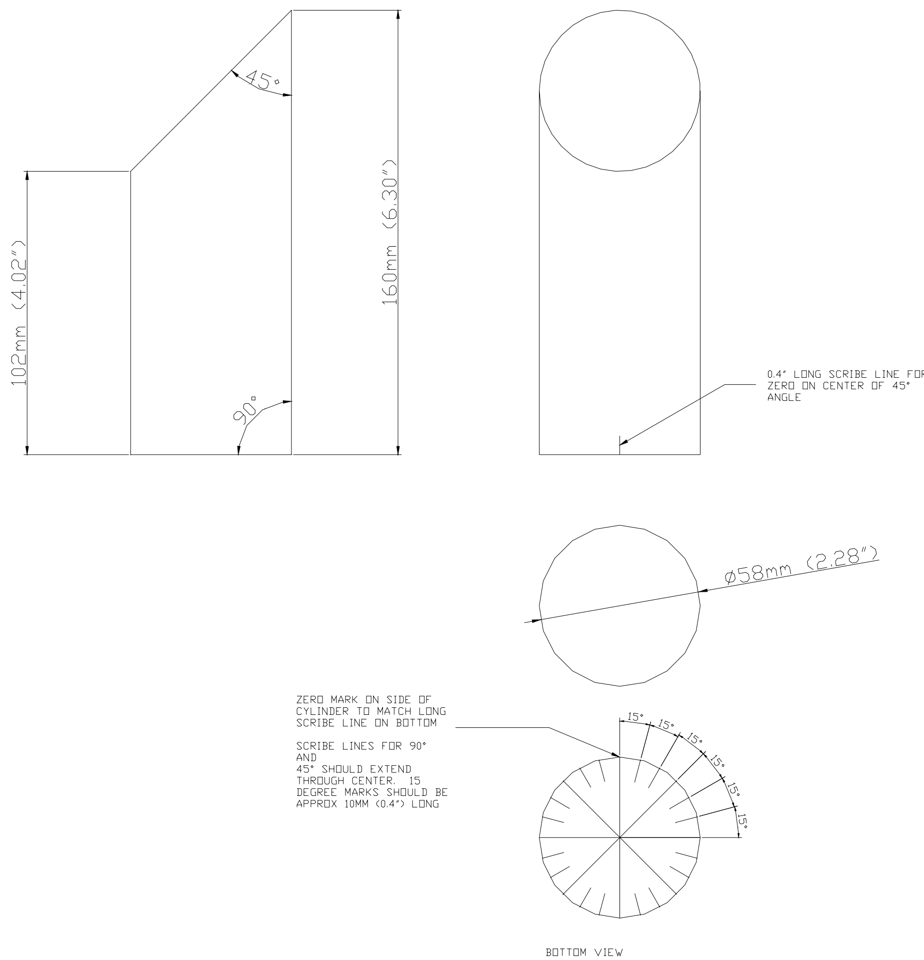
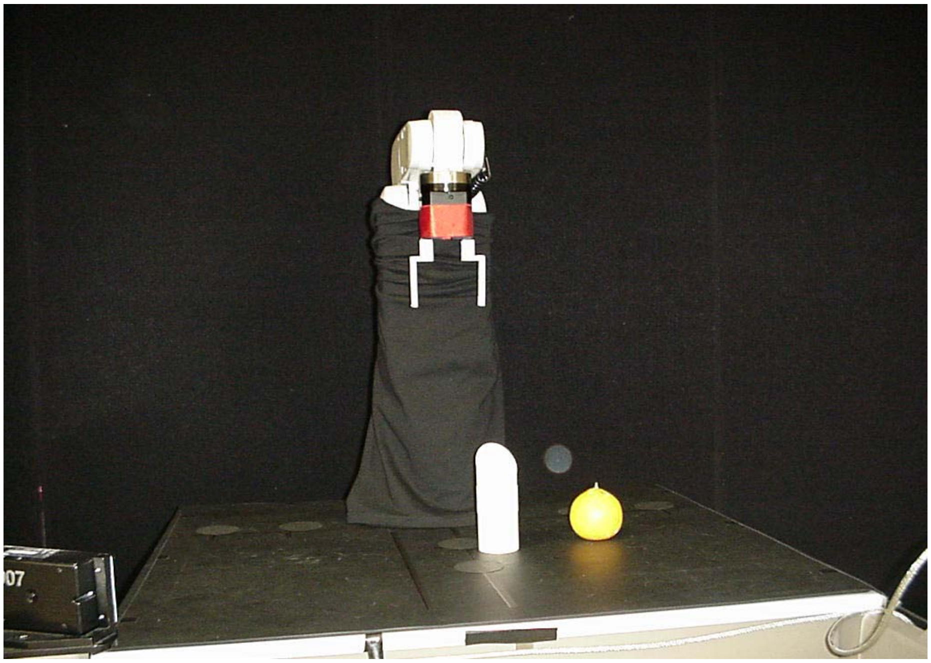


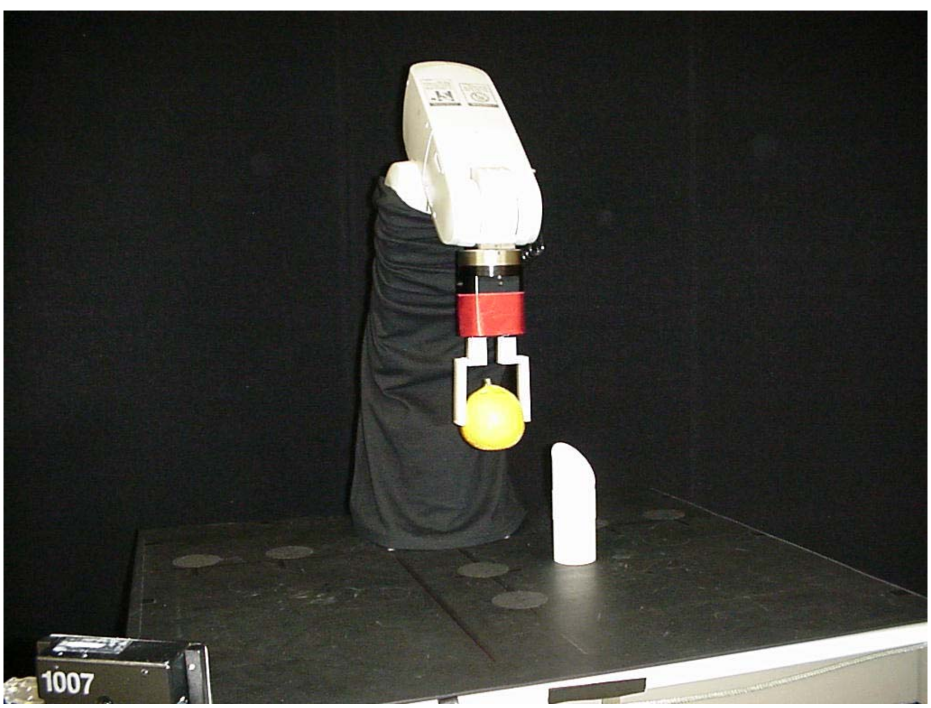
FIGURE 3
Test Object



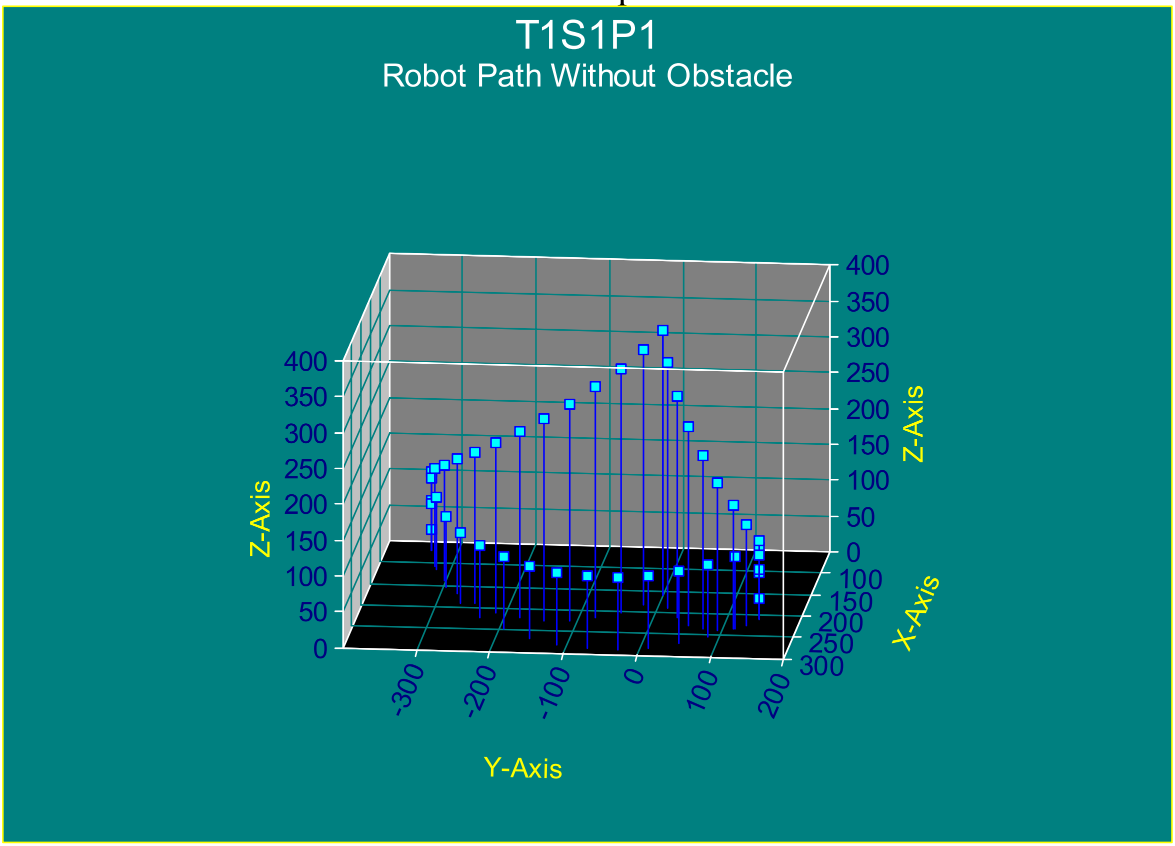
Photograph 1 Test Setup - Single Object Avoidance Test

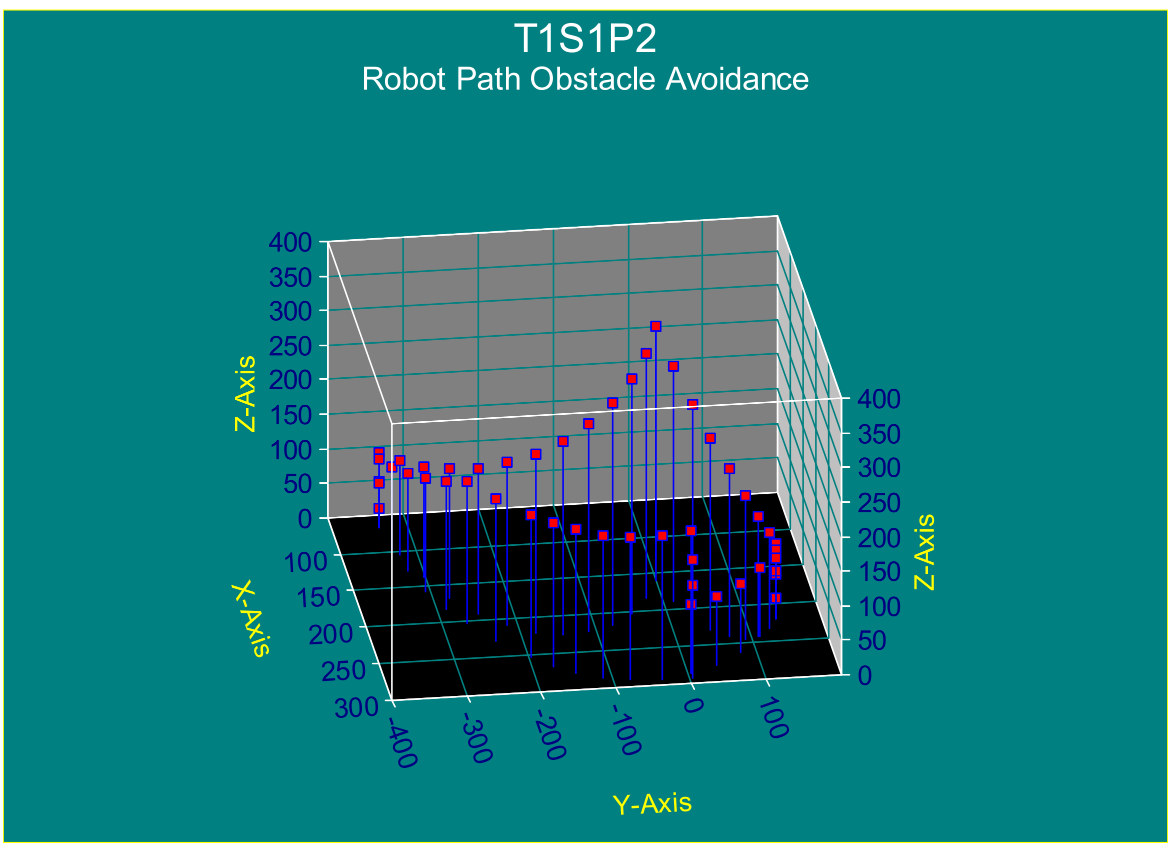


Photograph 2 Robot in Operation - Single Object Avoidance Test

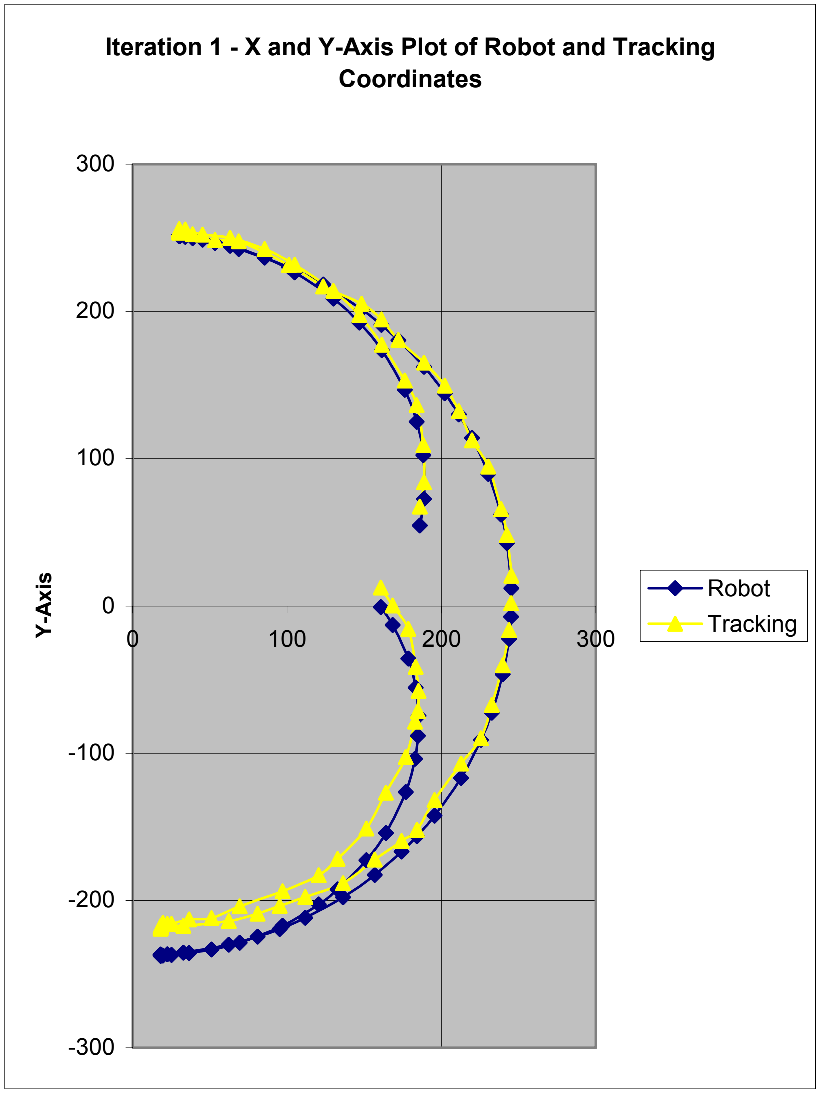


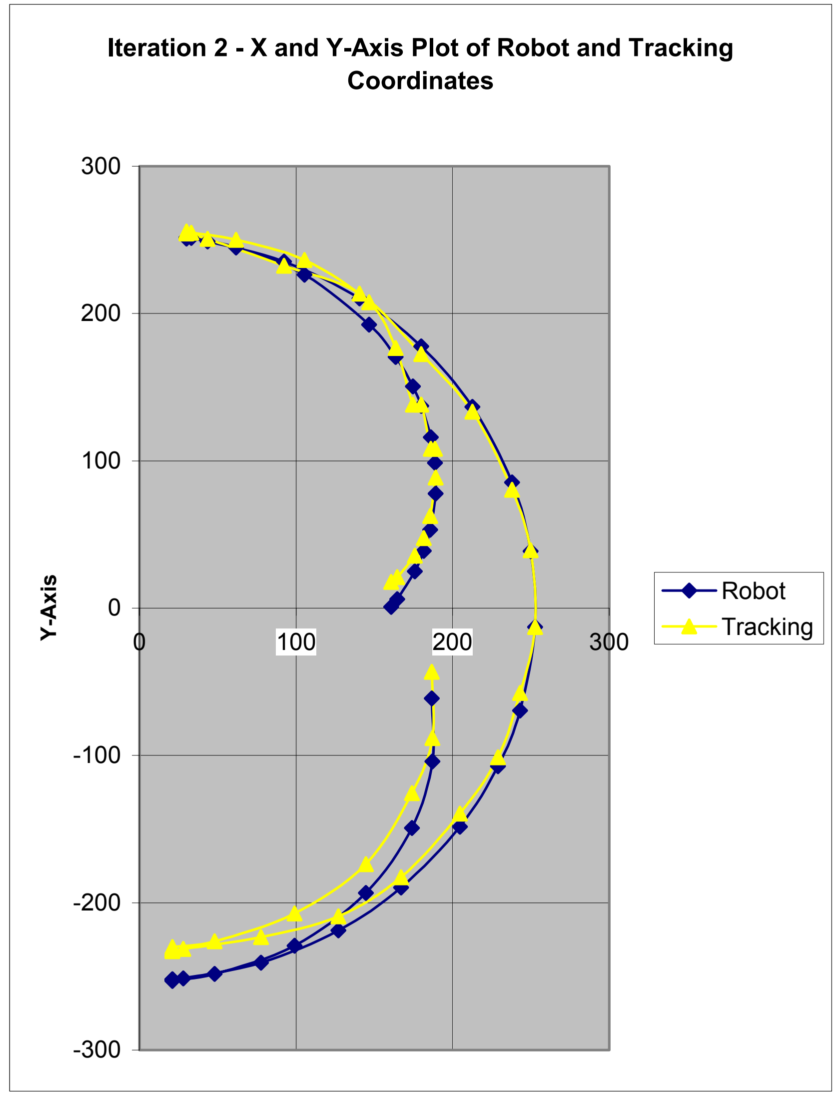
Test Five - Graph Set One





Test Six – X-Y Graph[h One





Appendix A

Physical Layout of REFES System and Mitsubishi RV1A Robotic Arm

A Mitsubishi RV1A robot is mounted at the rear center of a table measuring approximately three feet wide by two feet deep. The background, tabletop, and the robot arm with the exception of the grip are black or draped in black. Two stationary cameras (motion detection cameras) are mounted one each at the two front corners of the table. The cameras are mounted approximately one foot or less away from the table and approximately six inches above the table and are angled roughly toward the center of the robot base. The REFES system (VZX Imaging system) is mounted directly above the stationary camera on the right side looking toward the workstation. Above the REFES system is a stripe projection source. A computer monitor, keyboard, mouse, and computer are located on a table to the right of the workstation and are used to operate the REFES system.

The robot workspace on the tabletop is a semicircular band approximately 180 degrees around the origin of the robot coordinate and centered on the origin of the x-axis. The inside radius of the band is approximately 204 millimeters and the outside radius is approximately 417 millimeters as measured from the 0,0 point of the robot (center of J1 axis) (see fig. 1).

Appendix B

Physical Layout of REFES and Staubli Robotic Arm

A Staubli robot is mounted suspended above the rear center of a table measuring approximately 991 mm by 813 mm deep. The background, tabletop, and the robot arm with the exception of the grip are black or draped in black. The grip of the robot is a scissors type of claw with the opposing legs of the claw curved towards each other to facilitate the grasping a cylindrical object. Opposed across the table and facing the robot arm two stationary motion detection cameras are mounted one each at the two front corners of the table. The cameras are mounted approximately one foot or less away from the table and approximately six inches above the table and are angled roughly toward the center of the robot claw. A third (VZX camera system) is mounted directly above the stationary camera on the right side looking toward the robot claw. Above the REFES is a stripe projection source. The worktable surface is illustrated in figure 1.
A flat black matte board with location target circles drawn on the board to locate and orient the cylindrical test object was affixed to the worktable surface with double sided tape (see figure 1). The orientation lines on the target circles were oriented to correspond to the x and y-axis of the robot. Only locations that exhibited the highest accuracy were used.

Robot / Test Object Coordinates

The Staubli robot does not readily have fiduciary measuring surfaces external to the robot. The end effector mount and the location of the end effector claw are known by experimentation. X and y-axis coordinate measurements were taken by moving the test object to the center of each target location circle and recording the location coordinates from the robot controller display. The z-axis was constant for the object and was measured from the plane of the robot working-tab